(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 510 634 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
07.07.1999 Bulletin 1999/27

(51) Int Cl.6: **G06F 17/30**

(21) Application number: **92106939.9**

(22) Date of filing: **23.04.1992**

(54) Data base retrieval system

Datenbankauffindungssystem

Système de recouvrement pour base de données

(84) Designated Contracting States:
DE FR GB

(30) Priority:

25.04.1991	JP	12276691
24.12.1991	JP	35634891
26.12.1991	JP	35790091
27.12.1991	JP	35967591
07.02.1992	JP	5696492
07.02.1992	JP	5696592

(43) Date of publication of application:
28.10.1992 Bulletin 1992/44

(73) Proprietor: **NIPPON STEEL CORPORATION**
Tokyo 100-71 (JP)

(72) Inventor: **Takada, Hiroshi,**
c/o Nippon Steel Corporation
Sagamihara-shi, Kanagawa-ken (JP)

(74) Representative: **VOSSIUS & PARTNER**
Postfach 86 07 67
81634 München (DE)

(56) References cited:
WO-A-89/11699 **US-A- 4 839 853**

- **INFORMATION SYSTEMS, GB, vol. 12, no. 2, 1987, pages 151 - 156 GEBHART F. 'TEXT SIGNATURES BY SUPERIMPOSED CODING OF LETTER TRIPLETS AND QUADRUPLETS'**
- **THE CANADIAN JOURNAL FOR INFORMATION SCIENCE vol. 13, no. 3, December 1988, pages 79 - 89 NELSON M.J. 'Comparison of signature and inverted files'**
- **SOFTWARE PRACTICE & EXPERIENCE. vol. 18, no. 4, April 1988, CHICHESTER GB pages 387 - 393 OWOLABI O., MCGREGOR D.R. 'Fast Approximate String Matching'**

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The present invention relates to a data base retrieval system for extracting necessary information from a data base.

[0002] In an existing data base searching technique, keyword addition is generally used as a search space compression method. When the number of objective records is relatively small, a full-record search method can be used. For example, the Boyer-Moore method has been proposed as an efficient full-record search method. Furthermore, an index method for automatically extracting a keyword from a search object, and generating an index is also known.

[0003] The keyword search method suffers from the following drawbacks:

(1) A keyword must be added to each record;
(2) When arbitrary keywords are added, the number of keywords becomes very large, therefore, management using, e.g., a thesaurus is required, and considerable maintenance costs are required; and
(3) Since keywords to be added are not always proper, a search omission occurs.

[0004] More specifically, in the existing data base retrieval method, especially when the number of documents (i.e., the number of records) becomes very large, there is a tendency for performance not to be improved in proportion to required cost.

[0005] On the other hand, in a full-record search method, the above-mentioned problems are not posed. However, in an existing direct search method, when the number of records becomes very large, the search time considerably exceeds the interrogation time range, and is not practical. The full-record search method is based on complete coincidence, and cannot perform fuzzy coincidence searching. In the full-record search method based on the Boyer-Moore method, data other than a document (e.g., such as physical time-series data) cannot be processed.

[0006] As a method for performing full-record searching, a method disclosed in Japanese Patent Laid-Open No. 3-174652 is known. In this method, an index table, i.e., a character component table using entry characters as indices is formed in advance on the basis of search objective records, thereby narrowing the search range upon execution of full-record searching. However, since full-record searching is performed in the narrowed search range, the search time is long, and fuzzy coincidence searching cannot be performed.

[0007] Furthermore, the index method is suitable for documents such as English texts in which words are separated since the unit of information in such documents is a word. In this case, the index method requires some syntax analysis. The index method is not suitable for documents such as Japanese texts in which words are not separated. Furthermore, since a dictionary including all the possible sets of expressional variations of words must be formed, the system load is considerable.

[0008] Another method for retrieval of text documents involves the matching of search keys with documents using sub-strings or n-grams as disclosed in "Software Practice & Experience, vol. 18, no. 4, April 1988, Chichester (GB), pp. 387-93, Owolaibi O., McGregor D.R. "Fast Approximate String Matching". This method involves the extraction of n-grams from strings in a dictionary to form an n-gram table, and the comparison of n-grams extracted from an input string (search key) with the n-gram table of the dictionary to produce a matched set of strings. In forming the n-gram table, sub-strings of adjacent n characters are extracted from each string in the dictionary and statistics of occurrence frequency of the extracted sub-strings in the dictionary in the form of an n-gram table are taken. In the n-gram table each n-gram is accounted "1" if any string or dictionary vocabulary word contains the n-gram. Thus, the dictionary data is essentially abstracted one-dimensionally into vectors each having n-gram elements. Each n-gram is maintained as an inverted file comprising a one-dimensional bit-array format.

[0009] Japanese Patent Laid-Open No. 3-125263 discloses a search method using a plurality of continuous character strings as indices. However, this method also performs complete coincidence searching, and cannot perform incomplete coincidence searching (i.e., fuzzy coincidence searching).

[0010] Such a data base retrieval system is required to compress and decode data to decrease the volume of data to be searched and reduce the required memory capacity.

[0011] The Huffman method, the Shannon-Fano method, the Gilbert-Moore method, the run-length coding method, and the like are known as typical methods of compressing and decoding data. Japanese Patent Laid-Open No. 2-78323 discloses a technique using the Huffman method.

[0012] A method for fixing the size of all the records (e.g., an L-byte length) is known to attain high-speed data storage and reference (access) operations to a data base when data to be searched has a variable length. According to this method, when an n-th record is to be accessed, an n x L byte position from the start address of a file can be read, and the storage location can be designated at a high speed. However, in this method, since the record size is set to be constant, insignificant dummy characters must be added to data having a smaller length than the predetermined size, and the data size is undesirably increased.

[0013] In contrast to this, according to a method of continuously writing variable-length data in a storage medium, insignificant dummy characters need not be added, and it is not necessary to increase the data size. However, according to this method, since various data record sizes are used, the records must be referred to sequentially in an access mode, and the reference (storage) position cannot be immediately obtained. Therefore, the access speed is decreased.

[0014] As described above, the conventional variable-length data storage and reference methods suffer from at least one of two drawbacks, i.e., an increase in data size and a decrease in access speed.

[0015] The above-mentioned data base retrieval system checks whether or not records include a search key and lists as a search result data records including the search key.

[0016] The list of the search results is formed and preserved. However, when the number of records is large, or when the search results are sequentially preserved, since the volume of data preserved in the list is large, a memory device for storing the data requires a large memory capacity. Since a time required for forming the list of the search results is prolonged, search work efficiency deteriorates.

[0017] In the above-mentioned searching operation, when searching is performed using a conditional expression (searching expression) consisting of a plurality of search keys, the conditional expression is formed by the plurality of search keys, and searching is performed using the formed expression. For example, a conditional expression ((A or B or C) and D) is formed by keys A, B, C, and D, and full-record searching is performed using this expression.

[0018] However, since such a searching operation uses a conditional expression consisting of a plurality of keys, the search time is very long, and cost performance is low when a condition is not satisfied. When searching is performed using a similar conditional expression, e.g., a conditional expression ((A or B or C) and E) similar to the above-mentioned conditional expression, a partial logical condition (A or B or C) of searching that has already been calculated cannot be re-utilized and must be searched again, resulting in poor efficiency.

[0019] In consideration of the above situation, the present invention provides a data base retrieval system which can attain full-record searching, can remarkably shorten the search time, and can also attain fuzzy coincidence searching.

[0020] The present invention also provides a compression and decoding system which can compress, at high speed, integer character data which is to be preserved in the above-mentioned retrieval system, and is arranged in a monotonously increasing order (ascending order), and can reduce the required capacity of a memory used for storing compressed data.

[0021] Additionally, the present invention provides a variable-length data storage and reference system, which can decrease the data size of variable-length data to be preserved in the above-mentioned search system, and can increase the access speed.

[0022] The present invention provides a data base retrieval system which can shorten a time required for forming a list of search results and can decrease the required memory capacity of a memory device used in the above-mentioned retrieval system.

[0023] A data base retrieval system according to the present invention allows high-speed conditional searching regardless of the complexity of conditional expressions, and can re-utilize search results in the above-mentioned retrieval system.

[0024] A data base retrieval system according to the present invention provides a memory for storing vicinity feature values of records as search objects for the records. A searching device obtains matching degrees between a vicinity feature value of a search key and the vicinity feature values of the data records of the search object. Record numbers may be output by the searching device in descending order of the matching degrees.

[0025] According to the system of the present invention, phase information of data (position information indicating the position of a search key in a record) as a factor for prolonging the search time upon execution of full-record direct searching is abstracted by extracting vicinity feature values. The search time depends only on the length of search key information. Therefore, high-speed searching in which the search time does not depend on the data value can be performed. Since search results can be obtained as matching degrees (containing rates) of search keys in the records, a versatile search system can be realized independently of, e.g., syntax. In addition, fuzzy searching can be realized by referring to the matching degrees arranged in descending order. Text data, physical measurement data, signal waveform data, image data, acoustic data, and the like can be processed as a search object.

[0026] Upon compression and decoding of integer character data arranged in an ascending order, a divider divides the integer character data by a predetermined value. A quotient memory/comparison device compares a quotient obtained by the divider with a previously stored old quotient, and, when the obtained quotient is larger than the old quotient, provides an output corresponding to a difference between the new and old quotients. A memory stores a remainder obtained from the divider together with the difference between the two quotients when the difference is output from the quotient memory/comparison device. The memory stores only the remainder obtained from the divider when the quotient memory/comparison device does not output the difference between the two quotients (i.e., when the new and old quotients are equal). A decoder decodes original integer character data on the basis of the difference data between the two quotients and the remainder data stored in the memory.

[0027] According to the present invention, upon compression, data arranged in ascending order are divided, and the obtained quotient is compared with a previously obtained old quotient. The difference between the new and old quotients and the remainder are stored when a difference between the two quotients is detected. When no difference between the two quotients is detected, only the remainder data are stored. Therefore, since the amount of calculations can be greatly reduced compared to a conventional compression coding method, compression and decoding can be performed at high speed. Since no parameters such as statistical values associated with overall data are required, data can be easily added or deleted.

[0028] The system of the present invention provides a data memory for sequentially storing variable-length data, an ID assigning device for assigning an ID number to variable-length data stored in the data memory, and a storage location memory for storing a storage location of the variable length data in the data memory in correspondence with the ID number assigned by the ID assigning device.

[0029] According to the present invention, when data is stored, the ID number and storage location of the stored data are stored in the storage location memory. When data is to be accessed from the data memory, the storage location of the data is read out from the storage location memory to access the data. Therefore, since the data storage location can immediately be obtained from the storage location memory, the data memory can be accessed at a high speed. Since dummy data need not be added to the data to maintain data records having a fixed length, the volume of data to be stored can be reduced, and the required capacity of the storage medium can be decreased.

[0030] Furthermore, the data base retrieval system of the present invention for searching a data base in a fuzzy search mode provides a searching device for searching the data base using a search key to obtain matching degrees of the search key for all the records. A comparator compares the matching degrees of the search key of the records with a predetermined threshold value. A list preparing device prepares a list of records which are determined by the comparator to have matching degrees of the search key larger than the threshold value, and a record list memory stores the list of the records prepared by the list preparing device.

[0031] According to the present invention, when the data base is searched in a fuzzy search mode, the data base is searched using a search key to obtain matching degrees of the search key for all the records. The matching degrees of the search key of the data records are compared with the threshold value, and a list of search results of records, which have matching degrees of the search key larger than the predetermined threshold value, is formed, thus storing data.

[0032] Therefore, since the size of the list of the search results can be decreased, the capacity of the memory can be decreased and the search time can be shortened.

[0033] The data base retrieval system of the present invention provides a searching device for performing full-record searching under a predetermined condition, a search result memory for storing search results of the searching device, and a conditional searching device for performing conditional searching using the search results stored in the search result memory.

[0034] According to the present invention, when full-record searching is performed under a plurality of conditions, the searching device performs searching under a predetermined one of the plurality of conditions, and the search results are stored in the search result memory. The conditional searching device performs searching under complicated conditions using the search results. Therefore, since the search results based on the predetermined condition are stored, and searching under the complicated conditions is performed using the search results, partial search results can be re-utilized, and high-speed conditional searching can be realized.

[0035] Other objects and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings, wherein:

Fig. 1 is a block diagram illustrating a data base retrieval system according to the present invention;
Fig. 2 is a view for explaining quantization of vicinity information according to the present invention;
Fig. 3 is a view illustrating an information structure to be stored according to the present invention;
Fig. 4 is a view illustrating a vicinity feature value matrix;
Fig. 5 is a view illustrating the data architecture of a compressed vicinity feature value;
Fig. 6 is a block diagram illustrating a compression and decoding system according to the present invention;
Fig. 7 is a block diagram illustrating a variable-length data storage and reference system according to the present invention;
Fig. 8 is a block diagram illustrating a data base retrieval system according to the present invention; and
Fig. 9 is a block diagram illustrating a data base retrieval system according to the present invention.

[0036] Fig. 1 illustrates a pattern search system based on vicinity feature values according to the present invention. In this search system, vicinity feature value data obtained by abstracting all the phase information of events (information) from all the objective records are formed in advance, and full-record searching is performed for a group of the data. A searching algorithm consists of a studying step and a searching step. In the studying step, vicinity feature value matrices are formed as phase information of the data records. In the searching step, a matching calculation between a search key and the vicinity feature value matrix is performed, and appreciation results representing matching degrees (similarities) are obtained for the data records. The steps will be explained below.

(1) Studying Step

[0037] In Fig. 1, a search object 10 is text data in, e.g., Japanese, English, German, French, Hebrew, Russian, -or the like, or is quantized waveform numerical value data, a chemical structural formula, gene information, or the like. Such a search object is normalized by a normalizing device 32. In general, a search object is expressed as a sequence of minimum information units (e.g., characters such as English letters in a text, real number values at a given time in a numerical value chart, or the like). The search object is converted into an n-gradation integer sequence. This processing is called data normalization.

[0038] For example, English text data can be converted into the following 256-gradation numerical value expression by directly using the ASCII code table.

...This is a pen. ...

T	h	i	s		i	s		a		p	e	n	.
84	104	105	115	32	105	115	32	97	32	112	101	110	46

[0039] In the codes described above, "T" corresponds to "84, "h" corresponds to "104", and so on.

[0040] Normalized data 12 is convoluted in the form of a vicinity feature value matrix 14 by a search object vicinity feature value extractor 34. In this case, various formulas for extracting vicinity feature values are proposed. The formula influences the sharpness of searching (lack of over-detection).

[0041] Assume that a j-th data character of an i-th record of the text is represented by $C_{i,j}$. A quantization value x associated with $C_{i,j}$, and a quantization value y associated with k data characters in the vicinity of $C_{i,j}$ are obtained as follows. In this case, assume that there are n objective records, and quantization of the i-th record will be explained below. If a row of normalized numerical values 135, 64, 37, 71, 101,... is aligned in the i-th record, as shown in Fig. 2, the quantization value x associated with $C_{i,j}$ is given by:

$$x = f(C_{i,j}) \tag{1}$$

The quantization value y associated with the k data characters in the vicinity of $C_{i,j}$, is given by:

$$y = g\ (C_{i,j}, C_{i,j+1}, C_{i,j+2}, ..., C_{i,j+k}) \tag{2}$$

[0042] The function $f(C_{i,j})$ is the n-stage quantization function associated with $C_{i,j}$. More specifically, $f(C_{i,j})$ is a value obtained by performing a predetermined calculation for the j-th data character $C_{i,j}$ of the i-th record, and is expressed by an integer within a range between 1 and n. Therefore, a position in the x-direction in a matrix (coordinates) shown in Fig. 3 is determined within the range between I and n according to the obtained value x.

[0043] The function $g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, ..., C_{i,j+k})$ is the m-stage quantization function associated with the k data characters in the vicinity prior to $C_{i,j}$. More specifically, $g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, ..., C_{i,j+k})$ is a value obtained by performing a predetermined calculation for the j-th data character $C_{i,j}$ of the i-th record and a predetermined number of data in the vicinity of the data character $C_{i,j}$, and is expressed by an integer within a range between I and m. For example, as shown in Fig. 2, when the j-th data character $C_{i,j}$ = 135 and k = 3, data information 64, 37, and 71 following the data 135 is extracted as $C_{i,j+1}$, $C_{i,j+2}$, and $C_{i,j+3}$, and a predetermined calculation is performed for a correlation among these data and the data 135. When the j-th data character $C_{i,j}$ is the next data 64, data 37, 71, 101 following the data 64 is extracted as $C_{i,j+1}$, $C_{i,j+2}$, and $C_{i,j+3}$, and a predetermined calculation is performed for a correlation among these data and the data 64.

[0044] A position in the y-direction in the matrix (coordinates) shown in Fig. 3 is obtained within the range between 1 and m according to the value y obtained in this manner. Therefore, when x and y are obtained as described above, the position on the matrix (coordinates) shown in Fig. 2 can be determined.

[0045] In this system, each information record is stored as a set of a serial number i and a significance w(x,y,i) with respect to x and y obtained as described above. The significance w(x,y,i) is obtained by a predetermined calculation of data x, y, and i. In general, the value of the significance w(x,y,i) may be fixed to 1.

[0046] Data is stored in units of data $C_{i,j}$ obtained as described above on the basis of the values x and y, as indicated by bars in Fig. 3. More specifically, data as a set of the serial number i of a record and its significance w(x,y,i) is stored at the coordinate position defined by the values x and y of the data character $C_{i,j}$. In Fig. 3, the length of each bar is increased every time such data is stored. If the significance w(x,y,i) is set to be 1, only data of the serial number i of a record is stored at the coordinate position defined by the values x and y.

[0047] In place of the above-mentioned matrix (x,y), a matrix (x,y) given by the following equation may be used.

$$(x,y) = (f(c_i),g(c_i,c_{i+j})) \qquad (3)$$

where f( ) and g( ) are arbitrary integer functions. In this case, the domain of variability of c_i is equal to the ranges of f ( ) and g( ).

[0048] More specifically, if an i-th integer value in a record is represented by c_i, a calculation for setting an element value of the matrix given by equation (3) is repeated for j, and this processing is performed for all i's. Thus, the structural information of each record is convoluted into an n-th-order square matrix, as shown in Fig. 4. In this case, since each matrix element has only two values, the matrix can be sufficiently expressed by an n-th-order bit matrix, in practice. In this case of English text data in the above-mentioned 256-gradation numerical value expression, a vicinity feature value matrix is defined by 256 x 256.

[0049] The following calculation method will be exemplified for the above-mentioned English text data.
In equation (3),
if

$f: x \rightarrow x$
$g: (x,y) \rightarrow y$
$j = 1,2$

then, as for the first character "T" of "This", neighboring (j=1) and next neighboring (j=2) correlations (T → h) and (T → i) are respectively convoluted by two values, and 1 is set at a bit corresponding to an element given by:

$$(x,y) = (84,104),(84,105)$$

This operation is performed for the respective characters. At the end point of the above-mentioned record, only information relating to one vicinity character is convoluted (n → .).

[0050] A signature number of a record is added to the vicinity feature value matrix formed in this manner by a signature number adder 36, and the matrix is stored as a structure file in a memory 16.

(2) Searching Step

[0051] A search key is input from a search key input device 18. For example, "This is a pen." is used as a search key. Key information of the search key is normalized to an integer character by a normalizing device 38 based on the same normalizing method as that in the studying step, thus obtaining a normalizing key 20.

```
84| 104| 105| 115| 32| 105| 115| 32| 97| 32| 112| 101| 110| 46|
```

[0052] Then, a search key vicinity feature value extractor 40 forms a series of sets of x and y from the beginning of the normalized numerical value character corresponding to each record using the same vicinity feature value extraction formulas f( ) and g( ). Based on the series of sets of x and y, a searching device 42 calculates a containing rate w_k of the search key with respect to a record k by totaling $V(x_j,y_j,k)$ for j = 1 to m.

[0053] In this case, when a record information list has a significance for a record i, $V(x_j,y_j,k)$ is determined to be equal to the significance; otherwise, $V(x_j,y_j,k)$ is determined to be 0.

[0054] Therefore, when data (i.e., bar) is present at the position (x,y) in Fig. 3 corresponding to the sets of x and y of the numerical value character to be searched, its significance value is stored at a storage location of the serial number i of a record indicated by the data on a separately arranged memory.

[0055] Note that a search key may be applied to the vicinity feature value matrix corresponding to each record by the following equation (4) to perform structural appreciation of search key information.

$$\text{score} = \Sigma\Sigma p(f(c_i), g(c_i, c_{i+j})) \qquad (4)$$

(where the appreciation function p(x,y) assumes 1 when (x,y) in the matrix is non-zero, and assumes 0 when it is zero.)

[0056] More specifically, if a matrix element of each record corresponding to neighboring (j=1) and next neighboring (j=2) vicinity feature values for a character (i=1) of interest of a search key is 1, 1 is accumulated, and this processing is repeated for i characters in the search key. Equation (4) can be executed at very high speed by a logical calculation of AND, OR, addition and the like.

[0057] An appreciation result output device 44 divides a structural appreciation value "score" (matching degree) obtained for the data records with an appreciation value (the number of characters in the search key information - k or twice the number of characters) upon complete coincidence so as to obtain containing probabilities of the search key, thus obtaining a list 22 of appreciation results. Furthermore, a sorter 46 sorts the list 22 in descending order of the containing probabilities to obtain a sorted list 24.

[0058] The sorted list 24 corresponds to the search result. With reference to upper records in the list, names of records including the search key and having high probabilities can be detected. Since the containing probabilities can be obtained from all the complete and incomplete coincidence data, fuzzy coincidence searching can be performed.

[0059] If the search key is present in a given record in a given document in a complete coincidence state, p( ) in equation (4) becomes 1 for all the i's and j's.

[0060] Since full-record searching is performed for all the pieces of information of the search key, the probability of search omission can be substantially zero.

[0061] The appreciation time of the search key for one record depends only on the number of characters in the key, and does not depend on the size of the record. Therefore, searching can be performed at very high speed.

[0062] When a logical calculation is performed between the search result lists, search calculation processing such as AND, OR, and the like for search conditions can be executed at high speed.

[0063] The vicinity feature value extraction formula given by equation (3) can be variously modified. For example, if

$f: x \rightarrow x$
$g: (x,y) \rightarrow x\text{-}y$ (or $|x\text{-}y|$)

then a vicinity feature value matrix can be formed using the difference between neighboring and next neighboring characters (or the absolute value of the difference) as correlation information. Alternatively, individual character integer values of some character strings may be subjected to four-rule calculations to extract vicinity feature values.

[0064] The vicinity feature values need not always be extracted from all the data of the records. For example, vicinity feature values may be generated by excluding a specific one or more integer values in record data, integer values within a specific range, or a specific one or more bits in bytes constituting a data character. When a record is constituted by two-byte characters like a Japanese text, vicinity feature values may be extracted from, e.g., lower bytes while excluding upper bytes.

[0065] In the above-mentioned case, a matrix generated by the vicinity feature values is a 256th-order bit matrix, and this matrix corresponds to 8 kbytes. Therefore, in a data base in which each record has about 1 kbyte of data, the above-mentioned system is not an efficient system. Thus, data compression is performed by a data compression device 48 to decrease the necessary capacity of the memory 16.

[0066] Fig. 5 shows an example of a data compression method. In this example, record names 16a (signature codes) each having an element value = 1 are accumulated as a data row of 1 byte/records in units of elements of the 256th-order vicinity feature value matrix. Therefore, a record name having an element value = 0 is excluded as unnecessary data.

[0067] When the number of records exceeds 255, since the record name 16a cannot be expressed by one byte, only the lower one byte is accumulated. For example, when the number of records is 10,000, each record name is expressed by two bytes, and the lower byte of the two bytes is used. Every time a record name code exceeds 255, a marker 16b is inserted in a data row.

[0068] Upon searching, data rows of the structure files corresponding to the vicinity feature values of a search key are extracted, and an appearance frequency table divided into units of record names is formed. In this case, every time the marker 16b appears, 255 is added to the record name code. On the basis of the appearance frequency table formed in this manner, the appreciation result list 22 shown in Fig. 1 is obtained.

[0069] When the data sequence of a given record name code exceeds half of all the records, it is determined that the vicinity feature value matrix element is common to the respective records, and the element may be excluded.

[0070] In the above embodiment, the normalizing device 32, the search object vicinity feature value extractor 34, the signature number adder 36, the normalizing device 38, the searching device 42, the appreciation result output device 44, the sorter 46, and the data compression device 48 may be implemented by a computer program, but may also be implemented as a special-purpose hardware arrangement using logical circuit elements.

[0071] Fig. 6 shows an embodiment of a compression and decoding system according to the present invention. As shown in Fig. 6, integer character sequence data D1 is arranged in a monotonously increasing order (ascending order) such as "320, 333, 401,...". Each of these data elements is expressed by, e.g., 32 bits. The integer character sequence data D1 is supplied to a divider 52 in a compression apparatus. The divider 52 divides the input data by a predetermined value. In this embodiment, input data is divided by 255. The obtained quotient is supplied to a quotient memory and comparator 54, and the remainder is supplied to a compressed integer sequence D2 processor 56.

[0072] The quotient memory and comparator 54 compares a new quotient P_{new} received from divider 52 with a stored old quotient P_{old}. The initial value of the old quotient P_{old} is 0. When $P_{new} > P_{old}$ the quotient memory and comparator 54 supplies a mark character C indicating carry and the difference (P_{new} - P_{old}) between the two quotients to the compressed integer sequence D2 processor 56, and stores the new quotient P_{new} in place of the stored old quotient P_{old}. When this condition is not satisfied, the quotient memory and comparator 54 supplies no data to the compressed integer sequence D2 processor 56.

[0073] In this embodiment, when the first data element "320" is divided by 255, a quotient = 1 and a remainder = 65 are obtained. Since the initial value of the old quotient P_{old} is 0, $P_{new} > P_{old}$ is satisfied, and the quotient memory and comparator 54 supplies the mark character C indicating a carry and the difference "1" between the quotients to the compressed integer sequence D2 processor 56, and stores the new quotient "1" in place of the stored old quotient "0".

[0074] The compressed integer sequence D2 processor 56 stores the mark character C indicating a carry and the difference "1" between the quotients supplied from the quotient memory and comparator 54, and the remainder "65" supplied from the divider 52.

[0075] When "333" is supplied as the integer sequence character data D1, the divider 52 divides it by 255. In this case, the quotient = 1 and the remainder = 78. The quotient memory and comparator 54 compares the new quotient P_{new} with the previously stored old quotient P_{old}. In this case, since both P_{new} and P_{old} are "1", the above-mentioned condition $P_{new} > P_{old}$ is not satisfied. Therefore, only the remainder data is supplied from the divider 52 to the compressed integer sequence D2 processor 56.

[0076] When the above-mentioned operation is repeated, compressed data is sequentially supplied to the compressed integer sequence D2 processor 56. The compressed data is stored in a memory 58.

[0077] In decoding, the compressed data stored in the memory 58 is fetched by the compressed integer sequence D2 processor 56, and is read by a reader 62. When the mark character C indicating a carry appears in the compressed data, the reader 62 supplies data immediately after the mark character C to a bias data memory 64. The reader 62 supplies remainder data to an adder 66 independently of the presence/absence of appearance of the mark character C.

[0078] Since the first compressed data in this embodiment contains a mark character C indicating a carry, the immediately following data "1" is supplied to the bias data memory 64. The remainder data "65" is supplied to the adder 66.

[0079] The bias data memory 64 stores a value I based on the quotient, as shown in Fig. 6. Memory 64 receives from reader 62 data ΔP (i.e., the difference between the two quotients) immediately after the mark character C. Memory 64 adds a product $L \times \Delta P$ (where L is a divisor and ΔP is the difference of the new and old quotients) to the stored value I, and stores the sum as a new value I. Bias data memory 64 outputs the new I value to the adder 66. The initial value of I is 0.

[0080] In this embodiment, as described above, the divisor L is 255 and a "1" is supplied as the data ΔP immediately after the mark character C. Therefore, bias data memory 64 stores a value of "255" obtained by adding 255 x 1 to the initial value "0" of I, and provides it to the adder 66.

[0081] The adder 66 adds I supplied from the bias data memory 64 to the remainder supplied from the reader 62. In this case, the adder 66 adds "255" supplied from the bias data memory 64 and the remainder data "65" supplied from the reader 62, thereby obtaining decoded data "320". The obtained decoded data is stored in a decoded integer character sequence D3 memory 68, and is output as needed.

[0082] According to this embodiment, in the compression mode, ascending data is divided by the divisor L, and the obtained quotient is compared with a previously stored old quotient. When there is a difference between the old and new quotients, the difference and the remainder are stored. When there is no difference between the old and new quotients, only the remainder data is stored. Therefore, since the calculation amount can be greatly reduced as compared to a conventional compression coding method, compression and decoding can be performed at high speed. Since no parameters such as statistical values associated with overall data are required, data can be easily added or deleted.

[0083] This compression and coding system can be applied to data processing in the above-mentioned data retrieval system.

[0084] Fig. 7 illustrates an embodiment of a variable-length data storage and reference system according to the present invention. As illustrated in Fig. 7, when variable-length data is stored, data D1 is stored in a data memory 76 of a storage medium in the order of data A, data B, and data C. In the case shown in Fig. 7, the lengths of data A and B are respectively 100 and 40. Therefore, as shown in an ID location corresponding table 74, data A is stored at a storage location "0", data B is stored at a storage location "100", and data C is stored at a storage location "140". These locations are stored in the ID location corresponding table 74.

[0085] The data D1 is also supplied to an ID assigner 72, and is assigned with data serial numbers (IDs). The data serial numbers (IDs) are serial numbers assigned in correspondence with data, as shown in the ID location corresponding table 74. In this case, IDs "1", "2", and "3" are respectively assigned to the data A, B, and C. The assigned data IDs are supplied to and stored in the ID location corresponding table 74.

[0086] In this manner, data D1 is stored in the data memory 76 and the data IDs and corresponding data storage locations are stored in the ID location corresponding table 74.

[0087] When variable-length data 70 is referred to (or read out), data 71 corresponding to a reference request or its ID is supplied to the ID assigner 72, and the ID assigner 72 outputs the ID of this data. The data ID is supplied to the ID location corresponding table 74, and the table 74 outputs the corresponding storage location. Data is read out from the data memory 76 on the basis of the output storage location, and is stored in a temporary data memory 78. The data stored in the temporary data memory 78 is output to an output device (not shown) such as a CRT according to a request from an operator.

[0088] The ID assigner 72 and the temporary data memory 78 comprise storage media, which can be accessed at high speeds, and the ID location corresponding table 74 and the data memory 76 comprise storage media, which can be accessed at low speeds. Therefore, since the data memory 76 for storing data is an inexpensive storage medium which can be accessed at low speed, the capacity of the data memory 76 can be sufficiently large. Since the ID assigner 72 and the data temporary memory 78 are storage media which can be accessed at high speed, ID assignment upon data storage and reference of data read out from the data memory 76 and stored in the temporary data memory 78 can be performed at high speed.

[0089] According to this system, as described above, when variable-length data is stored, the data D1 is stored in the data memory 76, and the IDs assigned to the respective data, and the storage locations of data are stored in the ID location corresponding table 74. When data is referred to (or read out), data corresponding to a reference request is supplied to the ID assigner 72, and the ID assigner 72 outputs the ID of this data. The output ID is supplied to the ID location corresponding table 74. The table 74 outputs the storage location corresponding to the ID, and data is read out from the data memory 76 on the basis of the output storage location.

[0090] Therefore, since the data storage location is accessed using correspondence between the data ID and the data storage location stored in the ID location corresponding table 74, an access position to a record can immediately be obtained, and a data read (search) access can be performed at high speed.

[0091] When data is stored, it is not necessary to maintain constant record sizes or to add dummy data to the data records. Thus, it is possible to prevent the volume of data to be stored from being increased.

[0092] This variable-length data storage and reference system can be applied to storage and reference operations of data in the above-mentioned data retrieval system.

[0093] Fig. 8 illustrates another embodiment of a system according to the present invention. As shown in Fig. 8, this system includes a searching device 82. Searching device 82 performs full-record searching of a search object 10 as a data base using a conditional expression consisting of a predetermined search key input from a search key input device (not shown).

[0094] The search result 83 obtained by the searching device 82 consists of a record number and a significance (weight) of the record, i.e., a containing rate (matching degree) of a search key for the record, as shown in Fig. 8. In Fig. 8, since the significance data (containing rates) of records 1, 2, and 3 are respectively 0.4, 0.6, and 1.0, record 3 has the highest containing rate (matching degree) of the three records.

[0095] The search results 83 obtained by the searching device 82 are supplied to a containing rate comparator 84. The containing rate comparator 84 compares a significance W of each record supplied from the searching device 82 with a threshold value θ input from a threshold value input device (not shown) to check if $\theta \leq W$. If $\theta \leq W$ is satisfied, i. e., when the significance W of a record is equal to or larger than the threshold value θ, the record and its significance data W are supplied to a record list preparing device 86, and are used as data for preparing a record number list.

[0096] When $\theta \leq W$ is not satisfied, i.e., when the significance W of a record is smaller than the threshold value θ, the above-mentioned data is not used as data for preparing the record number list, and is not supplied to the record list preparing device 86.

[0097] The record list preparing device 86 prepares the record number list with containing rates of the search key using data supplied from the containing rate comparator 84 (i.e., the numbers of records whose significance data W are equal to or larger than the threshold value θ) and the significance data W. This list is constituted by the record numbers and significance data, as illustrated in Fig. 8.

[0098] The record number list data prepared by the record list preparing device 86 is supplied to a record list memory 88. The record list memory 88 stores the record lists prepared by the record list preparing device 86 such as list 1, list

2, ...

[0099] According to this system, for data consisting of a record number and significance data of the record, i.e., a containing rate (matching degree) of a search key for the record, the significance data W is compared with a threshold value θ, and a record number list is prepared using only data of records whose significance data W is equal to or larger than the threshold value θ. The list is stored in record list memory 88.

[0100] Therefore, since the size of the prepared record number list can be small, the list preparation time can be shortened, and search work efficiency can be improved. Since the list to be stored can be decreased in size, the required capacity of a memory 88 which stores the lists can be decreased.

[0101] The record number list prepared by the record list preparing device 86 may be supplied to a sorter 90, and after the list is sorted in the order of significance data W, the sorted list may be stored in the record list memory 88. Alternatively, data 83 output from the searching device 82 may be supplied to the sorter 90, and after the data is sorted in the order of significance data W, the sorted data may be supplied to the containing rate comparator 84.

[0102] As will be described later, when data output from the searching device 82 has already been sorted in the descending order of significance data W, if the significance data W becomes smaller than θ upon comparison with the threshold value in the containing rate comparator 84, the subsequent comparison can be omitted.

[0103] This search system can be applied to processing of the search result in the above-mentioned data search system.

[0104] Fig. 9 illustrates an embodiment of a system according to the present invention. As shown in Fig. 9, this system has a searching device 92, a search result list memory 94, and a record list conditional search device 96. The searching device 92 performs full-record searching of a search object 10 as a data base using a condition expression consisting of a predetermined search key input from a search key input device 98.

[0105] For example, when keys input from the search key input device 98 are A and B, as shown in Fig. 9, the searching device 92 performs searching using the conditional expressions A and B, and search results are stored in the search result list memory 94. In Fig. 9, the record number list (3, 5, 10, 20) is searched by the conditional expression A, and the record number list (5, 10, 30) is searched by the conditional expression B. These record number lists are stored in the search result list memory 94 as search results.

[0106] The record list conditional search device 96 performs searching using a further complicated conditional expression based on the results stored in the search result list memory 94. For example, when searching using a conditional expression (A or B) or (A and B) is performed using the search results obtained by the conditional expressions A and B, the record list conditional search device 96 reads out the search results obtained by the conditional expressions A and B from the search result list memory 94, and performs searching based on the conditional expression (A or B) or (A and B) using the search results read from the search result list memory 94.

[0107] In this embodiment, since the results sorted in the search result list memory 94 are the record number list (3, 5, 10, 20) searched by the conditional expression A, and the record number list (5, 10, 30) searched by the conditional expression B, as described above, when searching using the conditional expression (A or B) is performed, these record number lists are logically ORed to obtain a record number list (3, 5, 10, 20, 30). Similarly, when searching using the conditional expression (A and B) is performed, these record number lists are logically ANDed to obtain a record number list (5, 10). The obtained record number lists are stored in the search result list memory 94.

[0108] Therefore, by using these lists, the record list conditional search device 96 can similarly perform searching using, e.g., a conditional expression "(A or B or C)" or "((A or B or C) and E)".

[0109] According to this embodiment, the search results obtained by conditional expression consisting of predetermined search keys are stored in the search result list memory 94 as record number lists. When searching using a complicated conditional expression as a combination of the keys is performed, the conditional searching is performed using the stored record number lists.

[0110] Therefore, searching using a complicated conditional expression need not be performed for all the records. Therefore, the search time can be shortened. In addition, since searching is performed by re-utilizing partial conditional search results, search efficiency can be improved.

[0111] This search system can be applied to searching of conditional expressions in the above-mentioned data retrieval system.

[0112] The data base retrieval system of the present invention stores vicinity feature values of records as search objects in the data records, obtains matching degrees between a vicinity feature value of a search key and the vicinity feature values of the search objects for the records, and outputs record numbers in descending order of matching degrees.

[0113] Therefore, according to the present invention, since phase information of data (position information indicating the position of a search key in a record) as a factor for increasing the search time upon execution of full-record direct searching is abstracted by extracting vicinity feature values, the search time depends only on the length of search key information. Therefore, high-speed searching in which the search time does not depend on the data volume can be realized. Since search results are obtained as the matching degrees (containing probabilities) of a search key in units of records, a versatile retrieval system independent from, e.g., syntax can be realized. Since incomplete coincidence searching can be performed by referring to the matching degree in the descending order, fuzzy searching can be attained, and the system of the present invention is strong against noise on the search key.

Claims

1. A data base retrieval system for retrieving information from a search object of a data base in response to an input search key, said search object comprising one or more data records and said system comprising

   first extraction means (34) for extracting vicinity feature values from each data record of the search object, said vicinity feature values indicating a correspondence among data elements of the data record;
   storage means (16) for storing the vicinity feature values of each data record;
   second extraction means (40) for extracting vicinity feature values of the search key, said vicinity feature values indicating a correspondence among data elements of the search key; and
   search means (42, 44) for obtaining matching degrees for the data records indicating a degree of correspondence between the vicinity feature values of each data record stored in said storage means (16) and the vicinity feature values of the search key extracted by said second extraction means (40), said search means (42, 44) being configured to provide as search results of said data base retrieval system data record numbers or names and corresponding matching degrees for the data records, characterized in that said first extraction means (34) calculates a plurality of sets of at least two quantization values (f, g) as the vicinity feature values of each data record, and
   said second extraction means (40) calculates at least one set of at least two quantization values (f, g) as the vicinity feature values of said search key.

2. A system according to claim 1, wherein said search means (42, 44) provides said data record numbers or names and corresponding matching degrees in a descending numerical order of said matching degrees.

3. A system according to claim 1 or 2, wherein said first extraction means (34) extracts the vicinity feature values of the data record by a convolution calculation among data elements of the data record.

4. A system according to any of claims 1 to 3, wherein said second extraction means (40) extracts the vicinity feature values of the search key by a convolution calculation among data elements of the search key.

5. A system according to any of claims 1 to 4, wherein the vicinity feature values of each data record are calculated in a similar manner as the vicinity feature values of the search key.

6. A system according to any of claims 1 to 5, wherein said quantization values include x and y, the quantization value x being associated with a j-th data element $C_{i,j}$ in an i-th data record of the search object, the quantization value y being associated with k data elements $C_{i,j+1}$, $C_{i,j+2}$, ... $C_{i,j+k}$ in the vicinity of the data element $C_{i,j}$ and the quantization values x and y are obtained by:

   $$x = f(C_{i,j})$$

   $$y = g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, \dots C_{i,j+k})$$

   wherein f(A) and g(B) are functions of A and B, respectively, wherein i,j and k are integers, and wherein a data record number or name i is stored for each data record in said storage means in a location defined by the obtained values x and y.

7. A system according to claim 6, wherein the quantization value x is expressed by one of n-stage values.

8. A system according to any of claims 1 to 5, wherein said quantization values include x and y, the quantization values $x = f(c_i)$ and $y = g(c_i, c_{i+j})$ are given to i-th data element c_i of a data sequence of the data record of the search object, and data c_{i+j} (j = 1, 2,...) in the vicinity of the data element c_i, the quantization values are used as element numbers of a matrix, one element value (= 1) of two values (1, 0) is given to the element numbers, and

a bit matrix generated by the quantization values for all i's is used as the vicinity feature values.

9. A system according to claim 8, wherein the quantization values $x = f(c_i)$ and $y = g(c_i, c_{i+j})$ are given to i-th data element c_i of a data sequence of the search key, and data c_{i+j} (j = 1, 2,...) in the vicinity of the data element c_i, an element value of the corresponding element number of the bit matrix is checked using the quantization values of the search key as the element numbers of the matrix, when the element value is 1, +1 is counted in the matching degree, and the matching degree for all i's is obtained for each data record.

10. A system according to any of claims 1 to 9, further comprising means (32, 38) for converting data in the data records of the search object and the search key into integer data elements in which each data element is n-gradation data.

11. A system according to claim 10, wherein n=256.

12. A system according to claim 10 or 11, further comprising:

means (52) for dividing said integer data elements by a predetermined value, said integer data elements being arranged in an ascending order of value;
means (54) for comparing a new quotient obtained by said means (52) for dividing with a previously stored old quotient, and when the new quotient is larger than the old quotient, providing a difference between the new and old quotients;
a compressed data memory (56, 58), storing, when said means (54) for comparing provides a difference between the new and old quotients, a remainder obtained by said means (52) for dividing together with the difference between the new and old quotients, and storing, when said means (54) for comparing does not provide a difference between the new and old quotients, only the remainder obtained by said means (52) for dividing; and
a decoder (62, 64, 66) decoding original integer data elements on the basis of the difference data and the remainder data stored in said compressed data memory (56, 58).

13. A system according to claim 12, wherein when the new quotient is larger than the old quotient, said means (54) for comparing provides the difference between the new and old quotients together with a mark indicating a carry.

14. A system according to any of claims 1 to 13, further comprising:

sequential storage means (76) for sequentially storing variable-length data corresponding to data records of said search object;
identification assignment means (72) for assigning identification numbers to identify regions of variable-length data stored in said sequential storage means (76); and
data locating means (74) for storing the identification numbers assigned by said identification assignment means (72) together with the corresponding storage locations of the regions of variable-length data to which said identification numbers refer;
wherein when data is stored in said sequential storage means (76), the identification number and storage location of the corresponding data region is stored in said data locating means (74), and when data is to be read from said sequential storage means (76), the storage location of the corresponding data region is read from said data locating means (74) to access the appropriate region of said sequential storage means (76).

15. A system according to claim 14, further comprising means (78) for temporarily storing data read from said sequential storage means (76).

16. A system according to any of claims 1 to 15, further comprising:

a comparator (84) comparing the matching degrees obtained by said search means (42, 82) for each of the data records with a predetermined threshold value;
means (86) for preparing a list of data records which are determined by said comparator (84) to have matching degrees larger than the threshold value; and
means (88) for storing as search results the list of data records prepared by said means (86) for preparing a list.

17. A system according to claim 16, further comprising means (90) for sorting the data records in a descending nu-

merical order of the matching degrees.

18. A system according to any of claims 1 to 17, further comprising:

full-record search means (92) for performing full-record searching under predetermined conditions;
search result memory (94) storing search results of said means (92); and conditional search means (96) for performing conditional searching using the search results stored in said search result memory (94);
wherein said conditional search means (96) searches under conditions as a combination of the conditions used in the searching of said full-record search means (92), on the basis of the search results stored in said search result memory (94).

19. A system according to claim 18, further comprising means (98) for inputting the conditions.

20. A method for retrieving information from a search object of a data base in response to an input search key, said search object comprising one or more data records and said method comprising the steps of:

extracting vicinity feature values of each data record from the search object, said vicinity feature values indicating a correspondence among data elements of the data record;
storing the vicinity feature values of each data record;
extracting vicinity feature values of the search key, said vicinity feature values indicating a correspondence among data elements of the search key;
obtaining matching degrees for the data records indicating a degree of correspondence between the stored vicinity feature values of each data record and the vicinity feature values of the search key; and
providing as search results of the information retrieving method data record numbers or names and corresponding matching degrees for the data records,

characterized in that, in said step of extracting the vicinity feature values of the data records, a plurality of sets of at least two quantization values are calculated as the vicinity feature values of each data record, and in said step of extracting the vicinity feature values of the search key, at least one set of at least two quantization values is calculated as the vicinity feature values of said search key.

21. A method according to claim 20, wherein said data record numbers or names and corresponding matching degrees are provided in a descending numerical order of said matching degrees.

22. A method according to claim 20 or 21, wherein the vicinity feature values of the data record are extracted by a convolution calculation among data in the data record.

23. A method according to any of claims 20 to 22, wherein the vicinity feature values of the search key are extracted by a convolution calculation among data elements of the search key.

24. A method according to any of claims 20 to 23, wherein the vicinity feature values of each data record are calculated in a similar manner as the vicinity feature values of the search key.

25. A method according to any of claims 20 to 24, wherein said quantization values include x and y, the quantization value x being associated with a j-th data element $C_{i,j}$ in an i-th data record of the search object, the quantization value y being associated with k data elements $C_{i,j+1}$, $C_{i,j+2}$, ... $C_{i,j+k}$ in the vicinity of the data element $C_{i,j}$, and the quantization values x and y are obtained by:

$$x = f(C_{i,j})$$

$$y = g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, \cdots C_{i,j+k})$$

wherein f(A) and g(B) are functions of A and B, respectively, wherein i,j and k are integers, and wherein a data record number or name i is stored for each data record in a memory location defined by the obtained values x and y.

26. A method according to claim 25, wherein the quantization value x is expressed by one of n-stage values.

27. A method according to any of claims 20 to 24, wherein said quantization values include x and y, the quantization values $x = f(c_i)$ and $y = g(c_i, c_{i+j})$ are given to i-th data element c_i of a data sequence of the data record of the search object, and data c_{i+j} ($j = 1, 2, \ldots$) in the vicinity of the data element c_i, the quantization the values are used as element numbers of a matrix, one element value (= 1) of two values (1, 0) is given to the element numbers, and a bit matrix generated by quantization values for all i's is used as the vicinity feature values.

28. A method according to claim 27, wherein the quantization values $x = f(c_i)$ and $y = g(c_i, c_{i+j})$ are given to i-th data element c_i of a data sequence of the search key, and data c_{i+j} ($j = 1, 2, \ldots$) in the vicinity of the data element c_i, an element value of the corresponding element number of the bit matrix is checked using the quantization values of the search key as the element numbers of the matrix, when the element value is 1, +1 is counted in the matching degree, and the matching degree for all i's is obtained for each data record.

29. A method according to any of claims 20 to 28, further comprising the step of converting data in the data records of the search object and the search key into integer data elements in which each data element is n-gradation data.

30. A method according to claim 29, wherein n=256.

31. A method according to claim 29 or 30, further comprising the steps of:

dividing said integer data elements by a predetermined value, said integer data elements being arranged in an ascending order of value;
comparing a new quotient obtained by said dividing step with a previously stored old quotient;
when the new quotient is larger than the old quotient, providing a difference between the new and old quotients and storing a remainder obtained by said dividing step together with the difference between the new and old quotients;
when the new quotient is not larger than the old quotient, storing only the remainder obtained by said dividing step; and
decoding original integer data elements on the basis of the stored difference data and remainder data.

32. A method according to claim 31, further comprising the step of storing the difference between the new and old quotients together with a mark indicating a carry when the new quotient is larger than the old quotient.

33. A method according to any of claims 20 to 32, further comprising the steps of:

sequentially storing in a variable-length data memory means (76) variable-length data corresponding to data records of said search object;
assigning identification numbers to the stored variable-length data;
storing in a storage location memory means (74) the assigned identification numbers in correspondence with storage locations of the variable-length data stored in said variable-length data memory means (76); and
when data is to be read from said variable-length data memory means (76), reading the storage location of the data from said storage location memory means (74) to access the appropriate storage location in said variable-length data memory means (76).

34. A method according to claim 33, further comprising the step of temporarily storing data read from said variable-length data memory means (76).

35. A method according to any of claims 20 to 34, further comprising the steps of:

comparing the matching degrees obtained by said obtaining step for each of the data records with a predetermined threshold value;
preparing a list of data records which are determined by said comparing step to have matching degrees larger than the threshold value; and
storing as search results the list of data records prepared by said list preparing step.

36. A method according to claim 35, further comprising the step of sorting the data records in a descending numerical order of the matching degrees.

37. A method according to any of claims 20 to 36, further comprising the steps of:

performing full-record searching under predetermined conditions;
storing search results of said full-record searching step; and
performing conditional searching using the search results stored in said search result storing step.

Patentansprüche

1. Datenbankauffindungssystem zum Auffinden von Informationen aus einem Suchobjekt einer Datenbank als Reaktion auf einen Eingabesuchschlüssel, wobei das Suchobjekt einen oder mehrere Datensätze aufweist und das System aufweist:

   eine erste Abfrageeinrichtung (34) zum Abfragen von Umgebungsmerkmalswerten aus jedem Datensatz des Suchobjekts; wobei die Umgebungsmerkmalswerte eine Entsprechung der Datenelemente des Datensatzes anzeigen;
   eine Speichereinrichtung (16) zum Speichern der Umgebungsmerkmalswerte jedes Datensatzes;
   eine zweite Abfrageeinrichtung (40) zum Abfragen von Umgebungsmerkmalswerten des Suchschlüssels, wobei die Umgebungsmerkmalswerte eine Entsprechung der Datenelemente des Suchschlüssels anzeigen; und
   eine Sucheinrichtung (42,44) zum Erhalten von Übereinstimmungsgraden für die Datensätze, die einen Grad der Entsprechung zwischen den Umgebungsmerkmalswerten jedes in der Speichereinrichtung (16) gespeicherten Datensatzes und den Umgebungsmerkmalswerten des durch die zweite Abfrageeinrichtung (40) abgefragten Suchschlüssels anzeigen, wobei die Sucheinrichtung (42,44) konfiguriert ist, um als Suchergebnisse des Datenbankauffindungssystems Datensatznummern oder -Namen und entsprechende Übereinstimmungsgrade für die Datensätze bereitzustellen,

   dadurch gekennzeichnet, daß die erste Abfrageeinrichtung (34) mehrere Sätze von mindestens zwei Quantisierungswerten (f,g) als die Umgebungsmerkmalswerte jedes Datensatzes berechnet, und
   die zweite Anfrageeinrichtung (40) mindestens einen Satz von mindestens zwei Quantisierungswerten (f,g) als die Umgebungsmerkmalswerte des Suchschlüssels berechnet.

2. System nach Anspruch 1, wobei die Sucheinrichtung (42,44) die Datensatznummern oder -Namen und entsprechenden Übereinstimmungsgrade in einer absteigenden numerischen Reihenfolge der Übereinstimmungsgrade bereitstellt.

3. System nach Anspruch 1 oder 2, wobei die erste Abfrageeinrichtung (34) die Umgebungsmerkmalswerte des Datensatzes durch eine Faltungsberechnung der Datenelemente des Datensatzes abfragt.

4. System nach einem der Ansprüche 1 bis 3, wobei die zweite Abfrageeinrichtung (40) die Umgebungsmerkmalswerte des Suchschlüssels durch eine Faltungsberechnung der Datenelemente des Suchschlüssels abfragt.

5. System nach einem der Ansprüche 1 bis 4, wobei die Umgebungsmerkmalswerte jedes Datensatzes in einer ähnlichen Weise wie die Umgebungsmerkmalswerte des Suchschlüssels berechnet werden.

6. System nach einem der Ansprüche 1 bis 5, wobei die Quantisierungswerte x und y aufweisen, wobei der Quantisierungswert x mit einem j-ten Datenelement $C_{i,j}$ in einem i-ten Datensatz des Suchobjekts verbunden ist, der Quantisierungswert y mit k Datenelementen $C_{i,j+1}$, $C_{i,j+2}$, ...$C_{i,j+k}$ in der Umgebung des Datenelements $C_{i,j}$ verbunden ist und die Quantisierungswerte x und y erhalten werden durch:

$$x = f(C_{i,j})$$

$$y = g\,(C_{i,j}, C_{i,j+1}, C_{i,j+2}, ..., C_{i,j+k})$$

   wobei f(A) und g(B) Funktionen von A bzw. B sind, wobei i, j und k Ganzzahlen sind und wobei eine Datensatznummer oder -Name i für jeden Datensatz in der Speichereinrichtung an einem Ort gespeichert wird, der durch die erhaltenen Werte x und y definiert wird.

7. System nach Anspruch 6, wobei der Quantisierungswert x durch einen von n-stufigen Werten ausgedrückt wird.

8. System nach einem der Ansprüche 1 bis 5, wobei die Quantisieungswerte x und y aufweisen, die Quantisierungswerte $x=f(c_i)$ und $y=g(c_i, c_{i+j})$ dem i-ten Datenelement c_i einer Datensequenz des Datensatzes des Suchobjekts und Daten c_{i+j} ($j = 1, 2, ...$) in der Umgebung des Datenelements ci gegeben werden, die Quantisierungswerte als Elementnummern einer Matrix verwendet werden, ein Elementwert (= 1) von zwei Werten (1, 0) den Elementnummern gegeben wird, und eine Bit-Matrix, die von den Quantisierungswerten für alle i's erzeugt wird, als die Umgebungsmerkmalswerte verwendet wird.

9. System nach Anspruch 8, wobei die Quantisierungswerte $x = f(c_i)$ und $y = g(c_i, c_{i+j})$ dem i-ten Datenelement c_i einer Datensequenz des Suchschlüssels und Daten c_{i+j} ($j = 1, 2, ...$) in der Umgebung des Datenelements c_i gegeben werden, ein Elementwert der entsprechenden Elementnummer der Bit-Matrix geprüft wird, wobei die Quantisierungswerte des Suchschlüssels als die Elementnummern der Matrix verwendet werden, wenn der Elementwert 1 beträgt, +1 im Übereinstimmungsgrad gezählt wird, und der Übereinstimmungsgrad für alle i's für jeden Datensatz erhalten wird.

10. System nach einem der Ansprüche 1 bis 9, das ferner eine Einrichtung (32,38) zum Umwandeln von Daten in den Datensätzen des Suchobjekts und des Suchschlüssels in Ganzzahl-Datenelemente aufweist, in welchen jedes Datenelement ein Datenwert mit n Abstufungen ist.

11. System nach Anspruch 10, wobei n = 256 ist.

12. System nach Anspruch 10 oder 11, das ferner aufweist:

eine Einrichtung (52) zum Dividieren der Ganzzahl-Datenelemente durch einen vorherbestimmten Wert, wobei die Ganzzahl-Datenelemente in einer absteigenden Reihenfolge des Wertes angeordnet sind;
eine Einrichtung (54) zum Vergleichen eines neuen Quotienten, der durch die Einrichtung (52) zum Dividieren erhalten wird, mit einem zuvor gespeicherten alten Quotienten, und wenn der neue Quotient größer ist als der alte Quotient, Bereitstellen einer Differenz zwischen den neuen und alten Quotienten;
einen Komprimiertdatenspeicher (56,58), der, wenn die Einrichtung (54) zum Vergleichen eine Differenz zwischen den neuen und alten Quotienten bereitstellt, einen Rest, der durch die Einrichtung (52) zum Dividieren erhalten wird, zusammen mit der Differenz zwischen dem neuen und dem alten Quotienten speichert, und der, wenn die Einrichtung (54) zum Vergleichen keine Differenz zwischen dem neuen und dem alten Quotienten bereitstellt, nur den durch die Einrichtung (52) zum Dividieren erhaltenen Rest speichert; und
einen Decodierer (62,64,66), der ursprüngliche Ganzzahl-Datenelemente auf der Basis der in dem Komprimiertdatenspeicher (56,58) gespeicherten Differenzdaten und der Restdaten decodiert.

13. System nach Anspruch 12, wobei, wenn der neue Quotient größer als der alte Quotient ist, die Einrichtung (54) zum Vergleichen die Differenz zwischen den neuen und alten Quotienten zusammen mit einer Markierung, die einen Übertrag anzeigt, bereitstellt.

14. System nach einem der Ansprüche 1 bis 13, das ferner aufweist:

eine sequentielle Speichereinrichtung (76) zum sequentiellen Speichern von Daten variabler Länge, die den Datensätzen des Suchobjekts entsprechen;
eine Identifikationszuweisungseinrichtung (72) zum Zuweisen von Identifikationsnummern, um Regionen von in der sequentiellen Speichereinrichtung (76) gespeicherten Daten variabler Länge zu identifizieren; und
eine Datenlokalisierungseinrichtung (74) zum Speichern der Identifikationsnummern, die durch die Identifikationszuweisungseinrichtungen (72) zugewiesen werden, zusammen mit den entsprechenden Speicherorten der Regionen von Daten variabler Länge, auf die die Identifikationsnummern verweisen;

wobei, wenn Daten in der sequentiellen Speichereinrichtung (76) gespeichert werden, die Identifikationsnummer und der Speicherort der entsprechenden Datenregion in der Datenlokalisierungseinrichtung (74) gespeichert werden, und, wenn Daten von der sequentiellen Speichereinrichtung (76) gelesen werden sollen, der Speicherort der entsprechenden Datenregion von der Datenlokalisierungseinrichtung (74) gelesen wird, um auf die passende Region der sequentiellen Speichereinrichtung (76) zuzugreifen.

15. System nach Anspruch 14, das ferner eine Einrichtung (78) zum vorübergehenden Speichern von Daten, die aus der sequentiellen Speichereinrichtung (76) gelesen wurden, aufweist.

16. System nach einem der Ansprüche 1 bis 15, das ferner aufweist:

einen Komparator (84), der die von der Sucheinrichtung (42,82) für jeden der Datensätze erhaltenen Übereinstimmungsgrade mit einem vorherbestimmten Schwellwert vergleicht;
eine Einrichtung (86) zum Vorbereiten einer Liste von Datensätzen, welche durch den Komparator (84) bestimmt werden, um Übereinstimmungsgrade aufzuweisen, die größer sind als der Schwellwert; und
eine Einrichtung (88)zum Speichern der Liste von Datensätzen, die von der Einrichtung (86) zum Vorbereiten einer Liste vorbereitet wird, als Suchergebnisse.

17. System nach Anspruch 16, das ferner eine Einrichtung (90) zum Sortieren der Datensätze in einer absteigenden numerischen Reihenfolge der Übereinstimmungsgrade aufweist.

18. System nach einem der Ansprüche 1 bis 17, das ferner aufweist:

eine Volldatensatz-Sucheinrichtung (92) zum Durchführen einer Volldatensatz-Suche unter vorherbestimmten Bedingungen;
einen Suchergebnisspeicher (94), der Suchergebnisse der Einrichtung (92) speichert; und
eine konditionale Sucheinrichtung (96) zum Durchführen konditionaler Suche unter Verwendung der in dem Suchergebnisspeicher (94) gespeicherten Suchergebnisse;
wobei die konditionalen Sucheinrichtung (96) unter Bedingungen sucht, wie einer Kombination der in der Suche der Volldatensatz-Sucheinrichtung (92) verwendeten Bedingungen, auf der Grundlage der in dem Suchergebnisspeicher (94) gespeicherten Suchergebnisse.

19. System nach Anspruch 18, das ferner eine Einrichtung (98) zum Eingeben der Bedingungen aufweist.

20. Verfahren zum Auffinden von Informationen in einem Suchobjekt einer Datenbank als Reaktion auf einen Eingabesuchschlüssel, wobei das Suchobjekt einen oder mehrere Datensätze aufweist und das Verfahren die Schritte aufweist:

Abfragen von Umgebungsmerkmalswerten jedes Datensatzes aus dem Suchobjekt, wobei die Umgebungsmerkmalswerte eine Entsprechung der Datenelemente des Datensatzes anzeigen;
Speichern der Umgebungsmerkmalswerte jedes Datensatzes;
Abfragen von Umgebungsmerkmalswerten des Suchschlüssels, wobei die Umgebungsmerkmalswerte eine Entsprechung der Datenelemente des Suchschlüssels anzeigen;
Erhalten von Übereinstimmungsgraden für die Datensätze, die einen Grad der Entsprechung zwischen den gespeicherten Umgebungsmerkmalswerten jedes Datensatzes und den Umgebungsmerkmalswerten des Suchschlüssels anzeigen; und
Bereitstellen von Datensatznummem oder -Namen und entsprechenden Übereinstimmungsgraden für die Datensätze als Suchergebnisse des Informationsauffindungsverfahrens,

dadurch gekennzeichnet, daß in dem Schritt des Abfragens der Umgebungsmerkmalswerte der Datensätze mehrere Sätze von mindestens zwei Quantisierungswerten als die Umgebungsmerkmalswerte jedes Datensatzes berechnet werden, und daß in dem Schritt des Abfragens der Umgebungsmerkmalswerte des Suchschlüssels mindestens ein Satz von mindestens zwei Quantisierungswerten als die Umgebungsmerkmalswerte des Suchschlüssels berechnet werden.

21. Verfahren nach Anspruch 20, wobei die Datensatznummern oder -Namen und entsprechenden Übereinstimmungsgrade in einer absteigenden numerischen Reihenfolge der Übereinstimmungsgrade bereitgestellt werden.

22. Verfahren nach Anspruch 20 oder 21, wobei die Umgebungsmerkmalswerte des Datensatzes durch eine Faltungsberechnung der Daten im Datensatz abgefragt werden.

23. Verfahren nach einem der Ansprüche 20 bis 22, wobei die Umgebungsmerkmalswerte des Suchschlüssels durch eine Faltungsberechnung der Datenelemente des Suchschlüssels abgefragt werden.

24. Verfahren nach einem der Ansprüche 20 bis 23, wobei die Umgebungsmerkmalswerte jedes Datensatzes in einer ähnlichen Weise wie die Umgebungsmerkmalswerte des Suchschlüssels berechnet werden.

25. Verfahren nach einem der Ansprüche 20 bis 24, wobei die Quantisierungswerte x und y aufweisen, der Quantisierungswert x mit einem j-ten Datenelement $C_{i,j}$ in einem i-ten Datensatz des Suchobjekts verbunden ist, der Quantisierungswert y mit k Datenelementen $C_{i,j+1}$, $C_{i,j+2}$, ..., $C_{i,j+k}$ in der Umgebung des Datenelements $C_{i,j}$ verbunden ist, und die Quantisierungswerte x und y erhalten werden durch:

$$x = f(C_{i,j})$$

$$y = g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, \ldots, C_{i,j+k}),$$

wobei f(A) und g(B) Funktionen von A bzw. B sind, wobei i, j und k Ganzzahlen sind, und wobei eine Datensatznummer oder -Name i für jeden Datensatz an einem Speicherort, der durch die erhaltenen Werte x und y definiert wird, gespeichert wird.

26. Verfahren nach Anspruch 25, wobei der Quantisierungswert x durch einen von n-stufigen Werten ausgedrückt wird.

27. Verfahren nach einem der Ansprüche 20 bis 24, wobei die Quantisierungswerte x und y aufweisen, die Quantisierungswerte $x = f(c_i)$ und $y = g(c_i, c_{i+j})$ dem i-ten Datenelement c_i einer Datensequenz des Datensatzes des Suchobjekts und Daten c_{i+j} (j = 1, 2,...) in der Umgebung des Datenelements c_i gegeben werden, die Quantisierungswerte als Elementnummern einer Matrix verwendet werden, ein Elementwert (= 1) von zwei Werten (1, 0) den Elementnummern gegeben wird, und eine Bitmatrix, die durch Quantisierungswerte für alle i's erzeugt wird, als die Umgebungsmerkmalswerte verwendet wird.

28. Verfahren nach Anspruch 27, wobei die Quantisierungswerte $x = f(c_i)$ und $y = g(c_i, c_{i+j})$ dem i-ten Datenelement c_i einer Datensequenz des Suchschlüssels und Daten c_{i+j} (j = 1, 2, ...) in der Umgebung des Datenelements c_i gegeben werden, ein Elementwert der entsprechenden Elementnummer der Bitmatrix geprüft wird, wobei die Quantisierungswerte des Suchschlüssels als die Elementnummern der Matrix verwendet werden, wenn der Elementwert 1 beträgt, +1 im Übereinstimmungsgrad gezählt wird, und der Übereinstimmungsgrad für alle i's für jeden Datensatz erhalten wird.

29. Verfahren nach einem der Ansprüche 20 bis 28, das ferner den Schritt des Umwandelns von Daten in den Datensätzen des Suchobjekts und des Suchschlüssels in Ganzzahl-Datenelemente, in welchen jedes Datenelement ein Datenwert mit n Abstufungen ist, aufweist.

30. Verfahren nach Anspruch 29, wobei n = 256 ist.

31. Verfahren nach Anspruch 29 oder 30, das ferner die Schritte aufweist:

Dividieren der Ganzzahl-Datenelemente durch einen vorherbestimmten Wert, wobei die Ganzzahl-Datenelemente in einer aufsteigenden Reihenfolge des Wertes angeordnet sind;

Vergleichen eines neuen Quotienten, der durch den Divisionsschritt erhalten wird, mit einem zuvor gespeicherten alten Quotienten;

wenn der neue Quotient größer als der alte Quotient ist, Bereitstellen einer Differenz zwischen dem neuen und dem alten Quotienten und Speichern eines Restes, der durch den Divisionsschritt erhalten wird, zusammen mit der Differenz zwischen den neuen und alten Quotienten;

wenn der neue Quotient nicht größer als der alte Quotient ist, Speichern nur des Restes, der durch den Divisionsschritt erhalten wird; und

Decodieren der ursprünglichen Ganzzahl-Datenelemente auf der Grundlage der gespeicherten Differenzdaten und der Restdaten.

32. Verfahren nach Anspruch 31, das ferner den Schritt des Speicherns der Differenz zwischen den neuen und alten Quotienten zusammen mit einer Markierung, die einen Übertrag anzeigt, wenn der neue Quotient größer ist als der alte Quotient, aufweist.

33. Verfahren nach einem der Ansprüche 20 bis 32, das ferner die Schritte aufweist:

sequentielles Speichern von Daten variabler Länge, die Datensätzen des Suchobjekts entsprechen, in einer

Speichereinrichtungfür Daten variabler Länge (76);
Zuweisen von Identifikationsnummern zu den gespeicherten Daten variabler Länge;
Speichern der zugewiesenen Identifikationsnummern in eine Speicherort-Speichereinrichtung (74) in Entsprechung mit Speicherorten der Daten variabler Länge, die in den Speichereinrichtung für Daten variabler Länge (76) gespeichert sind; und
wenn Daten von der Speichereinrichtung für Daten variabler Länge (76) gelesen werden sollen, Lesen des Speicherortes der Daten aus der Speicherort-Speichereinrichtung (74), um auf den passenden Speicherort in der Speichereinrichtung für Daten variabler Länge (76) zuzugreifen.

34. Verfahren nach Anspruch 33, das ferner den Schritt des vorübergehenden Speicherns von Daten, die aus der Speichereinrichtung für Daten variabler Länge (76) gelesen werden, aufweist.

35. Verfahren nach einem der Ansprüche 20 bis 34, das ferner die Schritte aufweist:

Vergleichen der Übereinstimmungsgrade, die durch den Erhaltungsschritt für jeden der Datensätze erhalten werden, mit einem vorherbestimmten Schwellwert;
Vorbereiten einer Liste von Datensätzen, welche dem Vergleichsschritt nach Übereinstimmungsgrade aufweisen, die größer sind als der Schwellwert; und
Speichern der Datensatzliste, die durch den Listenvorbereitungsschritt vorbereitet wird, als Suchergebnisse.

36. Verfahren nach Anspruch 35, das ferner den Schritt des Sortierens der Datensätze in einer absteigenden numerischen Reihenfolge der Übereinstimmungsgrade aufweist.

37. Verfahren nach einem der Ansprüche 20 bis 36, das ferner die Schritte aufweist:

Durchführen einer Volldatensatz-Suche unter vorherbestimmten Bedingungen;
Speichern der Suchergebnisse des Volldatensatz-Suchschrittes; und
Durchführen einer konditionalen Suche unter Verwendung der Suchergebnisse, die in dem Suchergebnis-Speicherschritt gespeichert werden.

Revendications

1. Système de recherche en base de données pour retrouver une information à partir d'un objet de recherche d'une base de données en réponse à une clé de recherche d'entrée, ledit objet de recherche comprenant un ou plusieurs enregistrements de données et ledit système comprenant :

un premier moyen d'extraction (34) pour extraire des valeurs de caractéristique de voisinage à partir de chaque enregistrement de données de l'objet de recherche, lesdites valeurs de caractéristique de voisinage indiquant une correspondance entre des éléments de données de l'enregistrement de données ;
un moyen de stockage (16) pour stocker les valeurs de caractéristique de voisinage de chaque enregistrement de données ;
un second moyen d'extraction (40) pour extraire des valeurs de caractéristique de voisinage de la clé de recherche, lesdites valeurs de caractéristique de voisinage indiquant une correspondance entre des éléments de données de la clé de recherche ;
un moyen de recherche (42, 44) pour obtenir des degrés de correspondance pour des enregistrements de données indiquant un degré de correspondance entre les valeurs de caractéristique de voisinage de chaque enregistrement de données stockées dans ledit moyen de stockage (16) et les valeurs de caractéristique de voisinage de la clé de recherche extraites au moyen dudit second moyen d'extraction (40), ledit moyen de recherche (42, 44) étant configuré pour produire en tant que résultats de recherche dudit système de recherche en base de données des numéros ou des noms d'enregistrement de données et des degrés de correspondance correspondants pour les enregistrements de données,

caractérisé en ce que ledit premier moyen d'extraction (34) calcule une pluralité de jeux d'au moins deux valeurs de quantification (f, g) en tant que valeurs de caractéristique de voisinage de chaque enregistrement de données et ledit second moyen d'extraction (40) calcule au moins un jeu d'au moins deux valeurs de quantification (f, g) en tant que valeurs de caractéristique de voisinage de ladite clé de recherche.

2. Système selon la revendication 1, dans lequel ledit moyen de recherche (42, 44) produit lesdits numéros ou noms d'enregistrement de données et lesdits degrés de correspondance correspondants selon un ordre numérique descendant desdits degrés de correspondance.

3. Système selon la revendication 1 ou 2, dans lequel ledit premier moyen d'extraction (34) extrait les valeurs de caractéristique de voisinage de l'enregistrement de données au moyen d'un calcul de convolution entre des éléments de données de l'enregistrement de données.

4. Système selon l'une quelconque des revendications 1 à 3, dans lequel ledit second moyen d'extraction (40) extrait les valeurs de caractéristique de voisinage de la clé de recherche au moyen d'un calcul de convolution entre des éléments de données de la clé de recherche.

5. Système selon l'une quelconque des revendications 1 à 4, dans lequel les valeurs de caractéristique de voisinage de chaque enregistrement de données sont calculées d'une manière similaire aux valeurs de caractéristique de voisinage de la clé de recherche.

6. Système selon l'une quelconque des revendications 1 à 5, dans lequel lesdites valeurs de quantification incluent x et y, la valeur de quantification x étant associée à un j-ième élément de donnée $C_{i,j}$ dans un i-ième enregistrement de données de l'objet de recherche, la valeur de quantification y étant associée à k éléments de données $C_{i,j+1}$, $C_{i,j+2}$, ... $C_{i,j+k}$ au voisinage de l'élément de données $C_{i,j}$ et les valeurs de quantification x et y sont obtenues comme suit :

$$x = f(C_{i,j})$$

$$y = g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, \ldots C_{i,j+k})$$

où f(A) et g(B) sont respectivement des fonctions de A et B où i, j et k sont des entiers et où un numéro ou nom d'enregistrement de données i est stocké pour chaque enregistrement de données dans ledit moyen de stockage en un emplacement défini par les valeurs obtenues x et y.

7. Système selon la revendication 6, dans lequel la valeur de quantification x est exprimée au moyen de l'une de valeurs de n étages.

8. Système selon l'une quelconque des revendications 1 à 5, dans lequel lesdites valeurs de auantification incluent x et y, les valeurs de quantification $x = f(c_i)$ et $y = g(c_i, c_{i+j})$ sont données au i-ième élément de données c_i d'une séquence de données de l'enregistrement de données de l'objet de recherche et des données c_{i+j} (j = 1, 2, ...) au voisinage de l'élément de données c_i, les valeurs de quantification sont utilisées en tant que numéros d'élément d'une matrice, une valeur d'élément (=1) de deux valeurs (1, 0) est donnée aux numéros d'élément, et une matrice de bits générée au moyen des valeurs de quantification pour tous les i est utilisée en tant que valeurs de caractéristique de voisinage.

9. Système selon la revendication 8, dans lequel les valeurs de quantification $x = f(c_i)$ et $y = g\ (c_i, c_{i+j}\ )$ sont données au i-ième élément de données c_i d'une séquence de données de la clé de recherche et des données c_{i+j} (j = 1, 2, ...) au voisinage de l'élément de données c_i, une valeur d'élément du numéro d'élément correspondant de la matrice de bits est vérifiée en utilisant les valeurs de quantification de la clé de recherche en tant que numéros d'élément de la matrice lorsque la valeur d'élément vaut 1, plus 1 est compté dans le degré de correspondance et le degré de correspondance pour tous les i est obtenu pour chaque enregistrement de données.

10. Système selon l'une quelconque des revendications 1 à 9, comprenant en outre un moyen (32, 38) pour convertir des données dans les enregistrements de données de l'objet de recherche et de la clé de recherche selon des éléments de données entiers dans lesquels chaque élément de données est une donnée de n gradations.

11. Système selon la revendication 10 dans lequel n = 256.

12. Système selon la revendication 10 ou 11, comprenant en outre :

un moyen (52) pour diviser lesdits éléments de données entiers par une valeur prédéterminée, lesdits éléments de données entiers étant agencés selon un ordre ascendant de valeurs ;
un moyen (54) pour comparer un nouveau quotient obtenu par ledit moyen (52) pour diviser avec un quotient ancien stocké préalablement et lorsque le nouveau quotient est supérieur à l'ancien quotient, pour produire une différence entre le nouveau quotient et l'ancien quotient ;
une mémoire de données comprimées (56, 58) qui stocke, lorsque ledit moyen (54) pour comparer produit une différence entre le nouveau quotient et l'ancien quotient, un reste obtenu par ledit moyen (52) pour diviser en association avec la différence entre le nouveau quotient et l'ancien quotient et pour stocker, lorsque ledit moyen (54) pour comparer ne produit pas une différence entre le nouveau quotient et l'ancien quotient, seulement le reste obtenu par ledit moyen (52) pour diviser ; et
un décodeur (62, 64, 66) qui décode des éléments de données entiers originaux sur la base des données de différence et des données de reste stockées dans ladite mémoire de données comprimées (56, 58).

13. Système selon la revendication 12, dans lequel, lorsque le nouveau quotient est supérieur à l'ancien quotient, ledit moyen (54) pour comparer produit la différence entre le nouveau quotient et l'ancien quotient en association avec un repère indiquant un report.

14. Système selon l'une quelconque des revendications 1 à 13, comprenant en outre :

un moyen de stockage séquentiel (76) pour stocker séquentiellement des données de longueur variable correspondant à des enregistrements de données dudit objet de recherche ;
un moyen d'assignation d'identification (72) pour assigner des numéros d'identification afin d'identifier des régions de données de longueur variables stockées dans ledit moyen de stockage séquentiel (76); et
un moyen de localisation de données (74) pour stocker les numéros d'identification assignés par ledit moyen d'assignation d'identification (72) en association avec les emplacements de stockage correspondants des régions de données de longueur variable auxquels lesdits numéros d'identification se réfèrent,

dans lequel, lorsque des données sont stockées dans ledit moyen de stockage séquentiel (76), le numéro d'identification et l'emplacement de stockage de la région de données correspondante sont stockés dans ledit moyen de localisation de données (74) et lorsque des données doivent être lues à partir dudit moyen de stockage séquentiel (76), l'emplacement de stockage de la région de données correspondante est lu à partir dudit moyen de localisation de données (74) afin d'accéder à la région appropriée dudit moyen de stockage séquentiel (76).

15. Système selon la revendication 14, comprenant en outre un moyen (78) pour stocker temporairement des données lues à partir dudit moyen de stockage séquentiel (76).

16. Système selon l'une quelconque des revendications 1 à 15, comprenant en outre :

un comparateur (84) qui compare les degrés de correspondance obtenus par ledit moyen de recherche (42, 82) pour chacun des enregistrements de données avec une valeur de seuil prédéterminée ;
un moyen (86) pour préparer une liste d'enregistrements de données qui sont déterminés par ledit comparateur (84) comme présentant des degrés de correspondance supérieurs à la valeur de seuil ; et
un moyen (88) pour stocker en tant que résultats de recherche la liste d'enregistrements de données préparée par ledit moyen (86) pour préparer une liste.

17. Système selon la revendication 16, comprenant en outre un moyen (90) pour trier les enregistrements de données selon un ordre numérique descendant des degrés de correspondance.

18. Système selon l'une quelconque des revendications 1 à 17, comprenant en outre :

un moyen de recherche d'enregistrement complet (92) pour réaliser une recherche d'enregistrement complet dans des conditions prédéterminées ;
une mémoire de résultat de recherche (94) qui stocke des résultats de recherche dudit moyen (92) ; et
un moyen de recherche conditionnelle (96) pour réaliser une recherche conditionnelle en utilisant les résultats de recherche stockés dans ladite mémoire de résultat de recherche (94),

dans lequel ledit moyen de recherche conditionnelle (96) réalise une recherche dans des conditions telles qu'une combinaison des conditions utilisées lors de la recherche dudit moyen de recherche d'enregistrement complet (92) sur la base des résultats de recherche stockés dans ladite mémoire de résultat de recherche (94).

19. Système selon la revendication 18, comprenant en outre un moyen (98) pour entrer les conditions.

20. Procédé de recherche d'une information à partir d'un objet de recherche d'une base de données en réponse à une clé de recherche d'entrée, ledit objet de recherche comprenant un ou plusieurs enregistrements de données et ledit procédé comprenant les étapes de :

extraction de valeurs de caractéristique de voisinage de chaque enregistrement de données à partir de l'objet de recherche, lesdites valeurs de caractéristique de voisinage indiquant une correspondance entre des éléments de données de l'enregistrement de données ;
stockage des valeurs de caractéristique de voisinage de chaque enregistrement;
extraction de valeurs de caractéristique de voisinage de la clé de recherche, lesdites valeurs de caractéristique de voisinage indiquant une correspondance entre des éléments de données de la clé de recherche ;
obtention de degrés de correspondance pour les enregistrements de données indiquant un degré de correspondance entre les valeurs de caractéristique de voisinage stockées de chaque enregistrement de données et les valeurs de caractéristique de voisinage de la clé de recherche ; et
production en tant que résultats de recherche dudit procédé de recherche d'une information des numéros ou noms d'enregistrement de donnés et de degrés de correspondance correspondants pour les enregistrements de donnés.

caractérisé en ce que, au niveau de ladite étape d'extraction des valeurs de caractéristique de voisinage des enregistrements de données, une pluralité de jeux d'au moins deux valeurs de quantification sont calculés en tant que valeurs de caractéristique de voisinage de chaque enregistrement de données et au niveau de ladite étape d'extraction des valeurs de caractéristique de voisinage de la clé de recherche, au moins un jeu d'au moins deux valeurs de quantification est calculé en tant que valeurs de caractéristique de voisinage de ladite clé de recherche.

21. Procédé selon la revendication 20, dans lequel lesdits numéros ou noms d'enregistrement de données et lesdits degrés de correspondance correspondants sont constitués selon un ordre numérique descendant desdits degrés de correspondance.

22. Procédé selon la revendication 20 ou 21, dans lequel les valeurs de caractéristique de voisinage de l'enregistrement de données sont extraites au moyen d'un calcul de convolution entre des données dans l'enregistrement de données.

23. Procédé selon l'une quelconque des revendications 20 à 22, dans lequel les valeurs de caractéristique de voisinage de la clé de recherche sont extraites au moyen d'un calcul de convolution entre des éléments de données de la clé de recherche.

24. Procédé selon l'une quelconque des revendications 20 à 23, dans lequel les valeurs de caractéristique de voisinage de chaque enregistrement de données sont calculées d'une manière similaire aux valeurs de caractéristique de voisinage de la clé de recherche.

25. Procédé selon l'une quelconque des revendications 20 à 24, dans lequel lesdites valeurs de quantification incluent x et y, la valeur de quantification x étant associée à un j-ième élément de données $C_{i,j}$ dans un i-ième enregistrementde donnés de l'objet de recherche, la valeur de quantification y étant associée à k éléments de donnés $C_{i,\,j+1}$, $C_{i,\,j+2}$, ... $C_{i,\,j+k}$ au voisinage de l'élément de données $C_{i,j}$ et les valeurs de quantification x et y sont obtenues comme suit :

$$x = f(C_{i,j})$$

$$y = g(C_{i,j}, C_{i,j+1}, C_{i,j+2}, \ldots C_{i,j+k}),$$

où f(A) et g(B) sont respectivement des fonctions de A et B, où i, j et k sont des entiers et où un numéro ou nom d'enregistrement de données i est stocké pour chaque enregistrement de données en un emplacement de mémoire défini par les valeurs obtenues x et y.

26. Procédé selon la revendication 25, dans lequel la valeur de quantification x est exprimée au moyen de l'une des valeurs de n étages.

27. Procédé selon l'une quelconque des revendications 20 à 24, dans lequel lesdites valeurs de quantification incluent x et y, les valeurs de quantification $x = f(c_i)$ et $y = g(c_i,c_{i+j})$ sont données au i-ième élément de données ci d'une séquence de données de l'enregistrement de données de l'objet de recherche et des données c_{i+j} (j = 1, 2, ...) au voisinage de l'élément de données c_i, les valeurs de quantification sont utilisées en tant que numéros d'élément d'une matrice, une valeur d'élément (=1) de deux valeurs (1, 0) est donnée aux numéros d'élément et une matrice de bits générée au moyen de valeurs de quantification pour tous les i est utilisée en tant que valeurs de caractéristique de voisinage.

28. Procédé selon la revendication 27, dans lequel les valeurs de quantification $x = f(c_i)$ et $y = g(c_i, c_{i+j})$ sont données au i-ième élément de données ci d'une séquence de données de la clé de recherche et des données c_{i+j} (j = 1, 2, ...) au voisinage de l'élément de données c_i, une valeur d'élément du numéro d'élément correspondant de la matrice de bits est vérifiée en utilisant les valeurs de quantification de la clé de recherche en tant que numéros d'élément de la matrice lorsque la valeur d'élément vaut 1, +1 est compté dans le degré de correspondance et le degré de correspondance pour tous les i est obtenu pour chaque enregistrement de données.

29. Procédé selon l'une quelconque des revendications 20 à 28, comprenant en outre l'étape de conversion des données dans les enregistrements de données de l'objet de recherche et de la clé de recherche selon des éléments de données entiers, dans lequel chaque élément de données est une donnée de n gradations.

30. Procédé selon la revendication 29, dans lequel n = 256.

31. Procédé selon la revendication 29 ou 30, comprenant en outre les étapes de :

division desdits éléments de données entiers par une valeur prédéterminée, lesdits éléments de données entiers étant agencés selon un ordre ascendant des valeurs ;
comparaison d'un nouveau quotient obtenu par ladite étape de division avec un ancien quotient stocké préalablement ;
lorsque le nouveau quotient est supérieur à l'ancien quotient, production d'une différence entre le nouveau quotient et l'ancien quotient et stockage d'un reste obtenu par ladite étape de division en association avec la différence entre le nouveau quotient et l'ancien quotient ;
lorsque le nouveau quotient n'est pas supérieur à l'ancien quotient, stockage de seulement le reste obtenu par ladite étape de division ; et
décodage d'éléments de données entiers originaux sur la base des données de différence et des données de reste stockées.

32. Procédé selon la revendication 31, comprenant en outre l'étape de stockage de la différence entre le nouveau quotient et l'ancien quotient en association avec un repère indiquant un report lorsque le nouveau quotient est supérieur à l'ancien quotient.

33. Procédé selon l'une quelconque des revendications 20 à 32, comprenant en outre les étapes de :

stockage séquentiel dans un moyen de mémoire de données de longueur variable (76) de données de longueur variable correspondant à des enregistrements de données dudit objet de recherche ;
assignation de numéros d'identification aux données de longueur variable stockées ;
stockage dans un moyen de mémoire d'emplacement de stockage (74) des numéros d'identification assignés en correspondance avec des emplacements de stockage des données de longueur variable stockées dans ledit moyen de mémoire de données de longueur variable (76); et
lorsque des données doivent être lues à partir dudit moyen de mémoire de données de longueur variable (76), lecture de l'emplacement de stockage des données à partir dudit moyen de mémoire d'emplacement de stockage (74) afin d'accéder à l'emplacement de stockage approprié dans ledit moyen de mémoire de données de longueur variable (76).

34. Procédé selon la revendication 33, comprenant en outre l'étape de stockage temporaire de données lues à partir dudit moyen de mémoire de données de longueur variable (76).

35. Procédé selon l'une quelconque des revendications 20 à 34, comprenant en outre les étapes de :

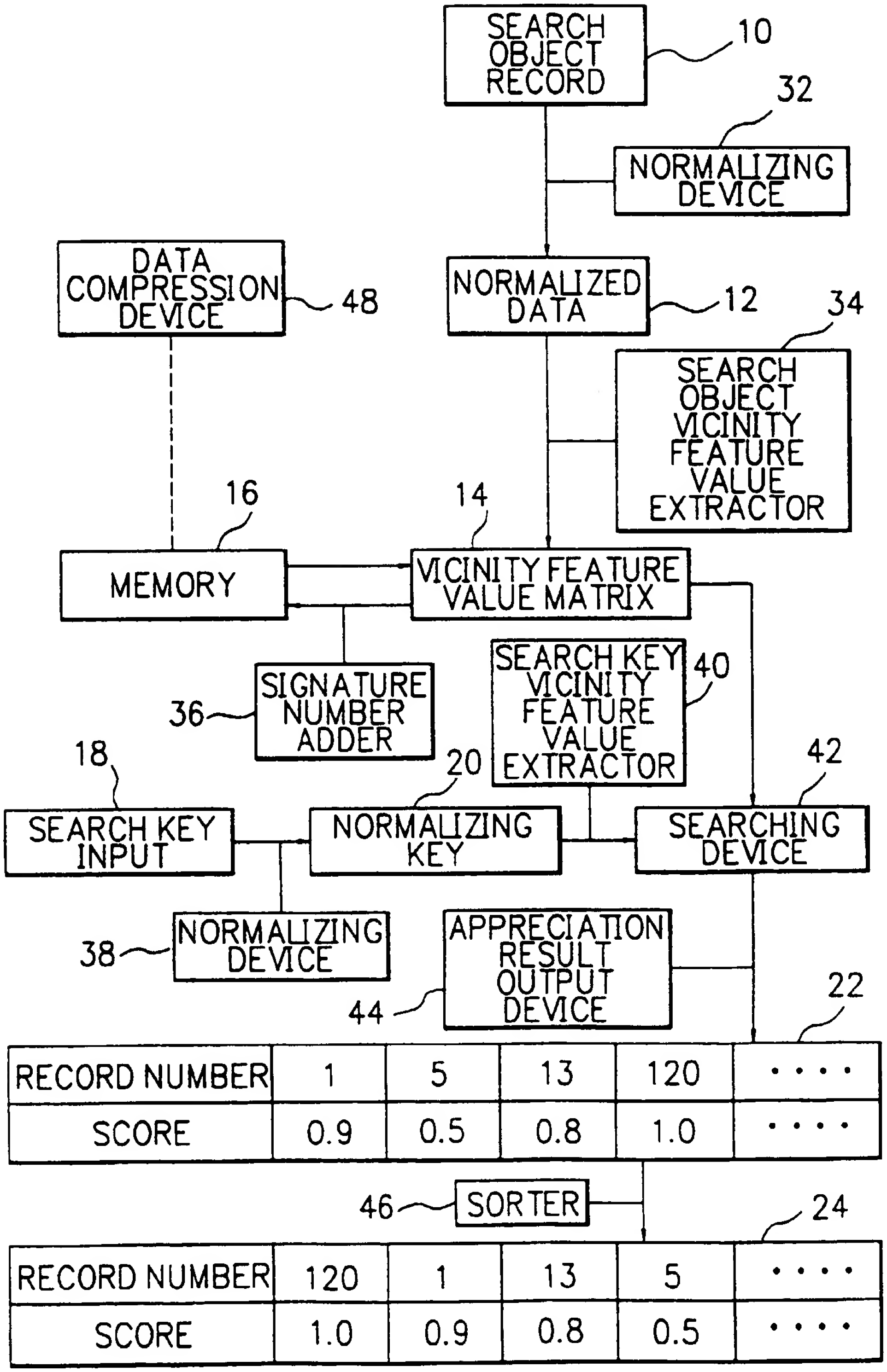
comparaison des degrés de correspondance obtenus par ladite étape d'obtention pour chacun des enregistrements de données avec une valeur de seuil prédéterminée ;
préparation d'une liste d'enregistrements de données qui sont déterminés par ladite étape de comparaison comme présentant des degrés de correspondance supérieurs à la valeur de seuil ; et
stockage en tant que résultats de recherche de la liste d'enregistrements de données préparée au moyen de ladite étape de préparation de liste.

36. Procédé selon la revendication 35, comprenant en outre l'étape de tri des enregistrements de données selon un ordre numérique descendant des degrés de correspondance.

37. Procédé selon l'une quelconque des revendications 20 à 36, comprenant en outre les étapes de :

réalisation d'une recherche d'enregistrement complet dans des conditions prédéterminées ;
stockage de résultats de recherche de ladite étape de recherche d'enregistrement complet; et
réalisation d'une recherche conditionnelle en utilisant les résultats de recherche stockés au niveau de ladite étape de stockage de résultat de recherche.

FIG. 1



RECORD NUMBER	1	5	13	120	• • • •
SCORE	0.9	0.5	0.8	1.0	• • • •

RECORD NUMBER	120	1	13	5	• • • •
SCORE	1.0	0.9	0.8	0.5	• • • •

FIG. 2

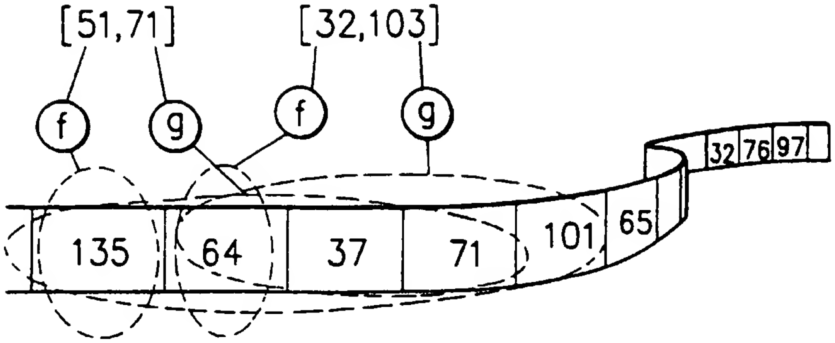

[51,71]
[32,103]
f
g
f
g
135
64
37
71
101
65
32
76
97

FIG. 3

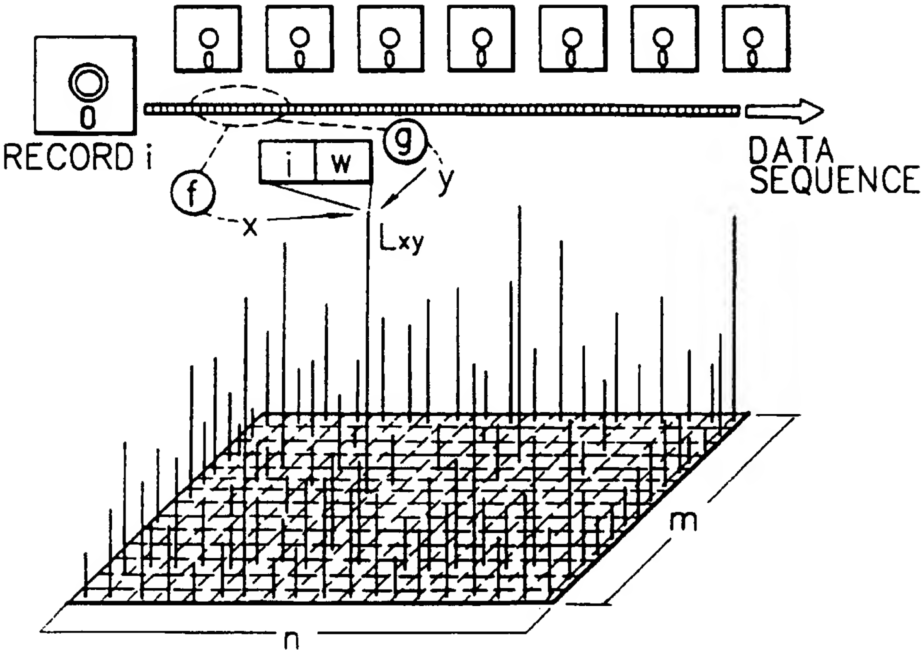

RECORD i
DATA SEQUENCE
f
g
i
w
x
y
Lxy
m
n

FIG. 4

```
                         n

     0 1 1 0 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0
     1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0
     1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 0 0
     1 0 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0
     0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 1
n    1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 1
     0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1
     1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1
                          .
                          .
                          .
     0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0
```

FIG. 5

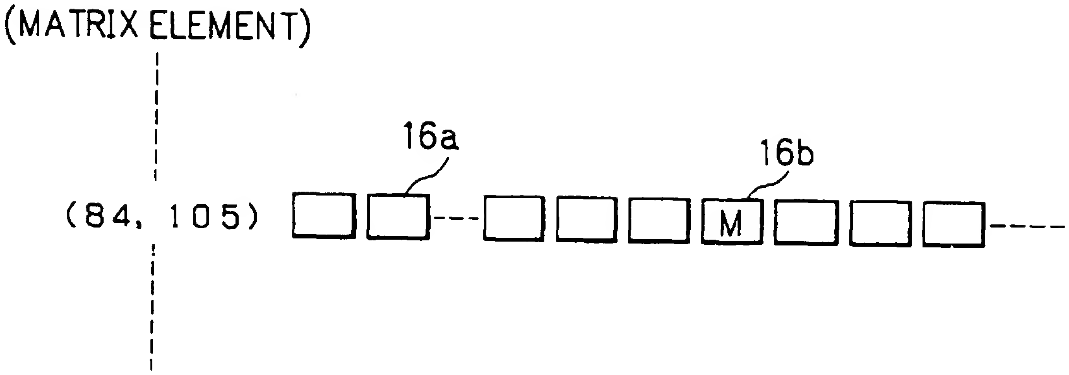
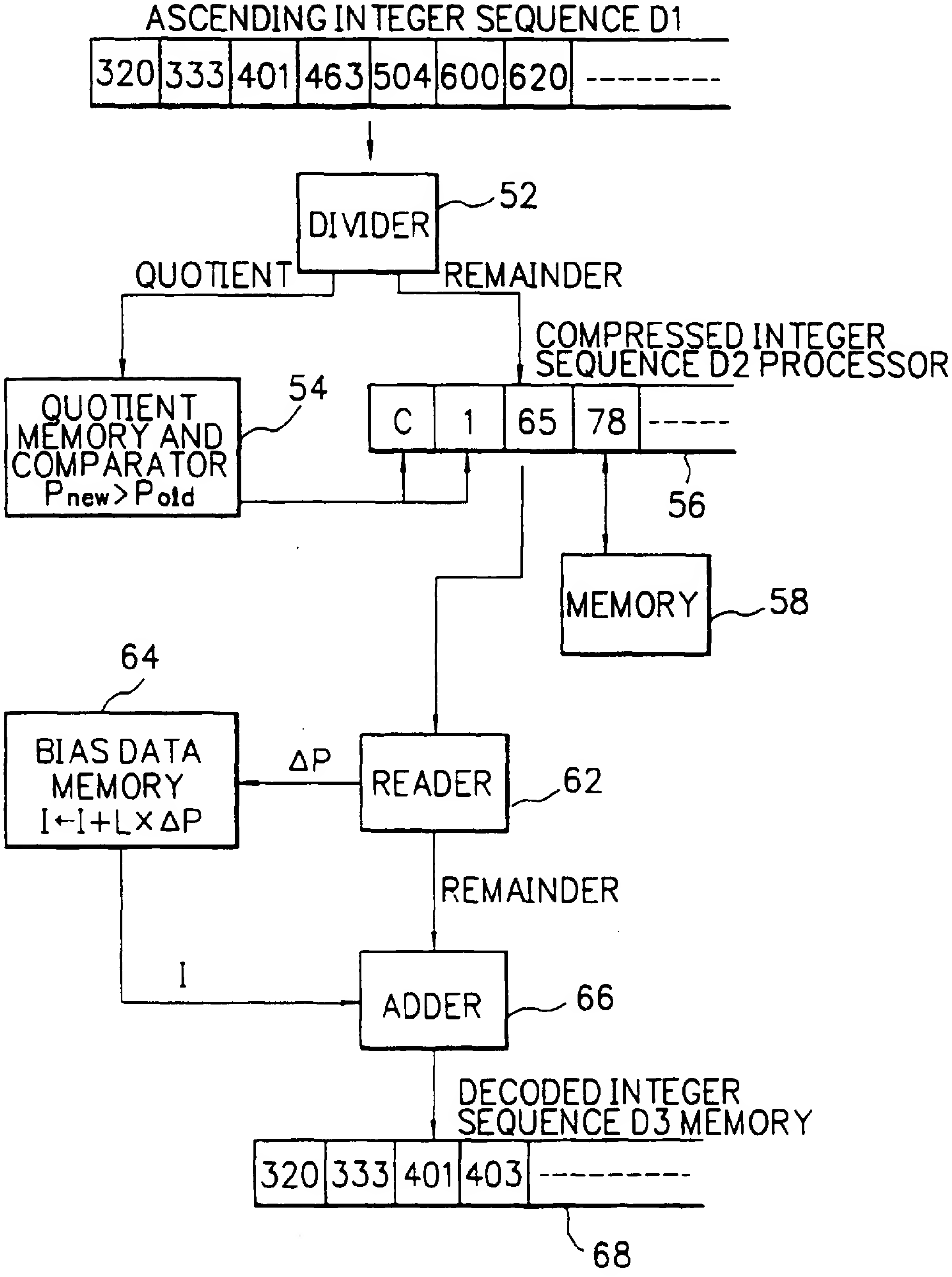


FIG. 6

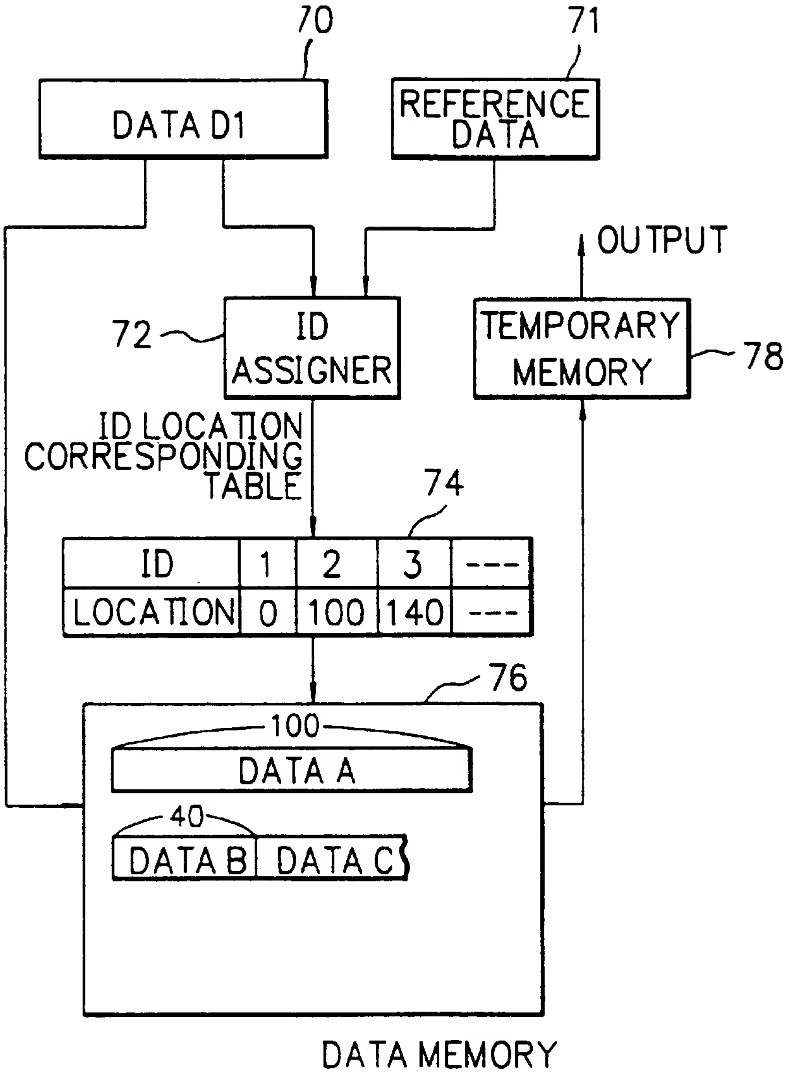

ASCENDING INTEGER SEQUENCE D1
320
333
401
463
504
600
620

DIVIDER
52
QUOTIENT
REMAINDER
COMPRESSED INTEGER SEQUENCE D2 PROCESSOR
QUOTIENT MEMORY AND COMPARATOR
Pnew>Pold
54
C
1
65
78

56
MEMORY
58
64
BIAS DATA MEMORY
I←I+L×ΔP
ΔP
READER
62
REMAINDER
I
ADDER
66
DECODED INTEGER SEQUENCE D3 MEMORY
320
333
401
403

68

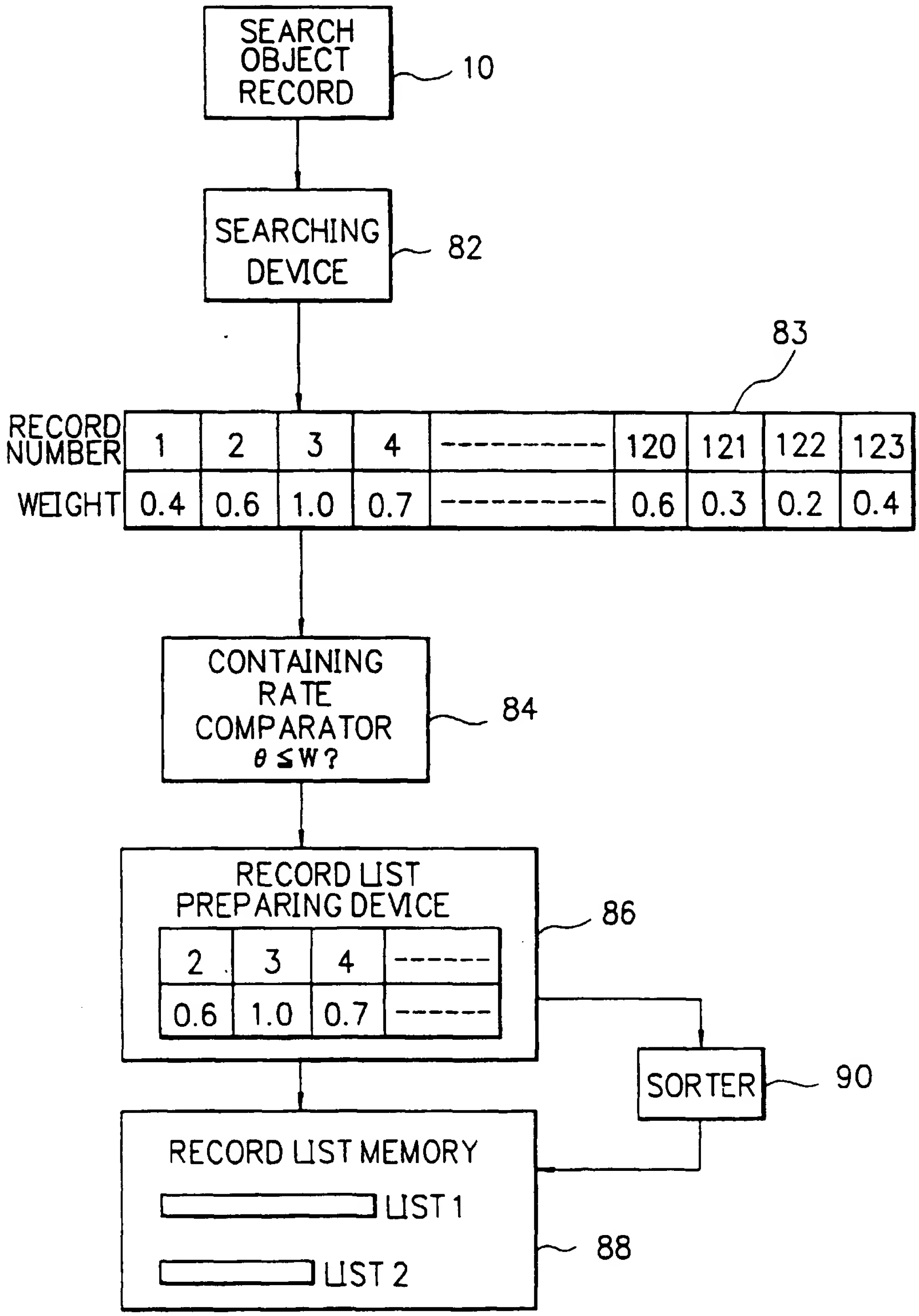
FIG. 7


70
71
DATA D1
REFERENCE DATA
OUTPUT
72
ID ASSIGNER
TEMPORARY MEMORY
78
ID LOCATION CORRESPONDING TABLE
74
ID
1
2
3

LOCATION
0
100
140

76
100
DATA A
40
DATA B
DATA C
DATA MEMORY

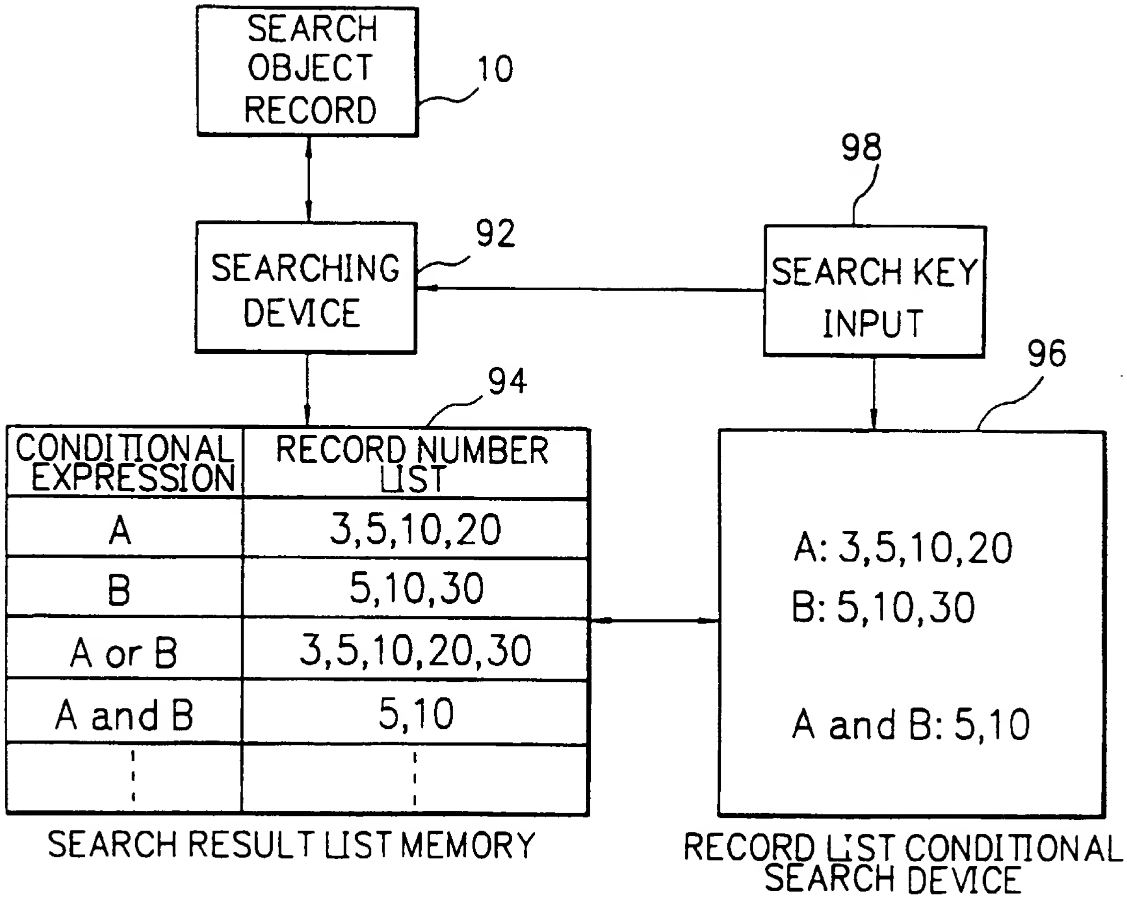
FIG. 8



RECORD NUMBER	1	2	3	4	----------	120	121	122	123
WEIGHT	0.4	0.6	1.0	0.7	----------	0.6	0.3	0.2	0.4

2	3	4	------
0.6	1.0	0.7	------

FIG. 9



CONDITIONAL EXPRESSION	RECORD NUMBER LIST
A	3,5,10,20
B	5,10,30
A or B	3,5,10,20,30
A and B	5,10
⋮	⋮